MSc thesis

Computer Science

Expert Performance in System Administration

Tuomas Husu

May 28, 2020

FACULTY OF SCIENCE
UNIVERSITY OF HELSINKI

Supervisor(s)

Prof. Antti Oulasvirta, Prof. Giulio Jacucci

Examiner(s)

Prof. Giulio Jacucci

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Faculty of Science	Computer Science

Tekijä — Författare — Author

Tuomas Husu

Työn nimi — Arbetets titel — Title

Expert Performance in System Administration

Ohjaajat — Handledare — Supervisors

Prof. Antti Oulasvirta, Prof. Giulio Jacucci

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
MSc thesis	May 28, 2020	85 pages, 14 appendice pages

Tiivistelmä — Referat — Abstract

System administration is a traditional and demanding profession in information technology that has gained little attention from human-computer interaction (HCI) research. System administrators operate in highly complex environment to keep business applications running and data available and safe.

In order to understand the essence of system administrators' skill, this thesis reports individual differences in 20 professional system administrators' task performance, task solutions, verbal reports, and learning histories. A set of representative tasks were designed to measure individual differences, and structured interviews were used to collect retrospective information about system administrators' skill acquisition and level of deliberate practice.

Based on the measured performance, the participants were divided into three performance groups. A group of five system administrators stood out from the 20 participants. They completed more tasks successfully, they were faster, they predicted their success more accurately, and they expressed more confidence during performance and anticipation. Although they had extensive professional experience, the study found no relationship between duration of experience and level of expertise.

The results are aligned with expert-performance research from other domains — the highest levels of performance in system administration are attained as a result of a systematic practice. This involves an investment of effort and makes the activity less enjoyable than competing activities. When studying the learning histories, the quantity and quality of the programming experience and other high-effort computer-related problem-solving activities were found to be the main differentiating factors between the 'expert' and less-accomplished participants.

ACM Computing Classification System (CCS)
Human-centered computing → Human-Computer Interaction (HCI) → Empirical studies in HCI

Avainsanat — Nyckelord — Keywords

System administrators, expert-performance approach, verbal protocols, human-computer interaction

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software Systems specialisation line

Contents

1 Introduction

This Master's thesis aims to shed light on what does it mean to be an expert system administrator. An *expert* is, according to Merriam-Webster (2020), someone "with the special skill or knowledge representing mastery of a particular subject", and *system administrator* is a traditional and common computer-related occupation, comprising about 5% of computer specialists in the United States (Bureau of Labor Statistics, U.S. Department of Labor, 2020). System administrators' duties are wide-ranging, but their primary responsibilities are the upkeep, configuration, and reliable operation of computer systems and networks (Nemeth et al., 2017).

1.1 Motivation

When system administrators succeed in their work, they are invisible — therefore, this user group has gained relatively little attention, although it is of great interest to information technology and management. System administrators have been said to be "the unsung heroes of the information age, working behind the scenes to configure, maintain, and troubleshoot the computer infrastructure that underlies much of modern life" (Takayama and Kandogan, 2006). System administrators can be thought of as the on-demand mechanics that keep business applications running and data available and safe. Although their role in modern technology-dependent organizations is decisive, individual differences in similar information technology occupations, such as software design, are reported to be significant (Mayer, 1997; Sonnentag et al., 2006). Nowadays, as the economy and daily life largely depends on information technology, the faultless running of systems and networks are of crucial relevance. It is essential to understand how expert performance in system administration is achieved. Scientific knowledge regarding skill in this group would be useful in education, training, management, recruitment, and user interface design.

1.2 Overview of the topic

The thesis studies system administrators to understand the essence of their skill. Their regular activities place very high demands on cognition, and complex problem-solving requires vast foreknowledge. The typical Linux system, for example, has around 6,000 built-in commands. Moreover, when solving problems, system administrators must be able to represent a multi-part computing environment—spanning systems and hardware, networks, applications, and software—with their settings and code. The equation also includes users who may misuse or overload the system, either accidentally or intentionally. System administrators' ability to anticipate potential outcomes of their actions is emphasized because feedback for success and errors can occur several steps further along in the process, if at all. Networked systems may fail unexpectedly, and finding the source of a problem is often more challenging than resolving it (Nemeth et al., 2017). System administrators must be constantly ready to learn about previously unknown system setups. Moreover, their tasks are often carried out under stress and external constraints. It is not sufficient that a solution just "works"; it must fulfill additional criteria like cybersecurity, maintainability, and ease of use.

However, instead the of integrated and visual environments available to other information technology professionals, system administrators often have to use multiple different user interfaces to solve a problem. These are typically command-line tools, text editors, and dialogue interfaces—each associated with a particular element and allowing only a limited "keyhole view" of a limited set of elements required for the solution (Limoncelli et al., 2007; Voronkov et al., 2019). When system administrators need to correct errors, there is nothing comparable to "undo" in graphical user interfaces or debugging tools available for programmers. Still, errors must be resolved across this keyhole view, one step at a time. Moreover, the objects that system administrators manipulate are sometimes only partially visible. They may be encapsulated within the software or reside remotely on other computers (and even other systems), and the interface may not necessarily make visible the constraints posed to those elements (Limoncelli et al., 2007). It is fair to state that, when it comes to cognitive complexity, system administrators' work is easily comparable to the work required by the most demanding professions in information technology, such as that of programmers and software architects.

To understand what is expertise, where it comes from, and whether it is possible to find objective measures for quantifying it in a complex domain of system administration, research needs to be conducted. Expertise research has a long tradition, but it has not been applied to system administration before. However, modern research techniques gives us tools to examine the essence of skill and its origin.

1.3 Research approach and questions

The goal of this thesis is to understand and account what, if anything, distinguishes outstanding individuals from less accomplished individuals in system administration. To answer this question, however, it is first necessary to understand the key mediating mechanisms of high performance. The thesis contributes a controlled experiment conducted with 20 professional system administrators to answer the following research questions derived from expertise research in other fields:

RQ1. Are some professional system administrators able to exhibit performance that is reliably superior to that of others?

RQ2. If individual differences in performance are observed, what are they?

RQ3. Is there a relationship between duration of professional system administration experience and measured performance?

RQ4. Is there a relationship between accumulated duration of practice and measured performance?

RQ5. Do the types of practice in which the professional system administrators have engaged differ as a function of performance?

RQ6. How much training, and what sort of training, is required by someone who hopes to become a high-performing system administrator?

To answer these questions, the *expert-performance approach* (Ericsson and Smith, 1991; Ericsson, Krampe, et al., 1993; Ericsson, 2004; Ericsson, 2006a; Ericsson, 2006b) is followed. The expert-performance approach involves designing a set of representative tasks to permit testing and measurement of individual differences in performance. Previous studies of skill in HCI have been criticized for the use of rather simple and unrepresentative tasks and materials, which may limit the generalizability and usefulness of results (Carroll, 1997; Mayer, 1997). For example, expert-novice studies of programming tended to use concise pieces of programming code to represent whole programs and conducted arbitrary comparisons between various types of code, such as "structured" versus "unorganized" code. In contrast, the primary goal in in this thesis is the designing of experimental tasks that are representative enough to warrant generalization to real-world use. The secondary aim is that tasks should be difficult enough to highlight differences among professional system administrators.

By collecting performance and process data, such as think-aloud protocols and solution paths, reliable differences in performance can then be related to differences in the mechanisms and cognitive processes mediating superior performance. To understand these differences further, the participants were also interviewed about their prior experience and their engagement in various related learning and problem-solving activities. A central hypothesis of this thesis is that the highest levels of performance in this domain are attained as a result of a regulated and systematic practice that involves an investment of effort, which makes this activity less enjoyable than competing activities. Lower levels of achievement in the same domain are hypothesized to be the result of repeatedly executing routine sequences of actions that lead to procedural memory and automatized actions. In contrast to automaticity, purposeful practice lead to cognitive restructuring with memory and attentional skills that enable flexibility and cognitive control of infrequent and non-routine behaviors. This could manifest in the transfer of problem-solving ability across tasks within this domain. Twenty professional system administrators participated in the experiment. Verbal think-aloud protocols were collected during the performance in all nine tasks. The protocol was designed to capture professional users' "thinking aloud" while they generated solutions to challenging tasks, and the verbal protocols were later analyzed to understand differences in problem-solving strategies and concepts. Users were also interviewed on their learning histories using a template derived from the deliberate practice framework.

1.4 Thesis structure

This thesis consists of six sections. Section 1 introduces the problem and gives a brief overview. Section 2 provides background and reviews the available literature, and Section 3 presents the experimental research method used in this thesis. Experiment results are presented in Section 4 and are discussed in Section 5. Finally, Section 6 concludes the thesis.

2 Background

The achievements of accomplished individuals are among the most admired and complex phenomena in our culture, and they offer several challenges to scientists attempting to study them. To better elucidate how expertise in system administration can be studied, this section reviews the existing literature and previous studies on expertise, expert performance, acquisition of superior performance, and findings from similar computer-related domains.

2.1 Expertise as innate talent

Some individuals are extremely good at what they do, be it in sports, musical performance, business, science, art, or medicine. When we encounter these top performers, we naturally tend to conclude that the individual was born with something extra. The idea of "a gift" in reference to ability dates back at least 2,700 years to the times of Odyssey.

> *Call in the inspired bard Demodocus.*
> *God has given the man the gift of song.* (Garvie et al., 1994)

The same themes can be seen in modern conversation when someone's "supernatural abilities" or "God-given gifts" are described. This phrasing suggests that their greatness was given to them, for reasons no one can explain, by someone or something apart from themselves. When Sloboda et al. (1996) studied the role of practice in the development of performing musicians, more than 75% of education professionals they interviewed believed that singing, composing, or playing instruments requires a special gift or talent.

The first studies of excellence were conducted in the late nineteenth century when Sir Francis Galton observed that the most valued achievements were made by members of a small number of eminent families (Galton, 1869). Galton found that "as the genetic bond to these families lessened, the likelihood that individuals had outstanding reputations also decreased" (Ericsson and Lehmann, 1999). In his research, Galton concluded that training is beneficial and required, but an individuals' performance is limited by a personal fixed upper bound for performance. The rare occurrence of expert performance could, therefore, be explained by innately talented individuals who were engaged in the domain and were endowed with sufficiently superior basic capacities.

In the late 1960s, Hungarian educators László and Klara Polgár decided to challenge the widespread assumption that hereditary factors such as gender and giftedness constitute an individual's chances of success (Ericsson, Prietula, et al., 2007). The Polgárs wanted to emphasize the power of education; they homeschooled their three daughters, and as part of their education, the girls started playing chess with their parents at a very young age. By the year 2000, all three daughters were ranked in the top ten female chess players in the world. The youngest daughter, Judit, became the youngest player ever to reach grandmaster status – and subsequently one of the world's top players defeating all the best male players (Ericsson, Prietula, et al., 2007; Flora, 2005).

Elite performers and their childhoods have been studied (Bloom, Sosniak, et al., 1985) and the findings are consistent with the Polgár sisters' story. All outstanding performers had practiced intensively, had studied with devoted teachers, and had been supported enthusiastically by their families throughout their developing years. However, according to Ericsson and Pool (2016), there is such a thing as "a gift." The real gift found in individuals is the ability to create – through the right sort of training and practice – abilities that they would not otherwise possess by taking advantage of the incredible adaptability of the human brain and body. Research during recent decades has shown that it is only the right sort of practice, carried out over a sufficient period of time, that leads to improvement.

2.2 Expertise as acquired knowledge and skill

Traditionally, professional expertise has been defined by the duration of experience and by the perceived mastery of knowledge and skill. In ancient Greek civilization, the primary factor influencing performance was considered to be accumulated experience, and age was believed to correlate with wisdom. Plato (ca. 370 B.C.E. / 2000) defined the ideal doctor as one with significant (and ideally, personal) experience of many diseases.

In pioneering research on expertise, expertise was typically determined by calculating total years of experience (e.g., Jeffries et al., 1981) and using peer-nomination procedures among highly experienced professionals (Elstein et al., 1978). Expert performance was considered an autonomous achievement that can be accomplished by a sufficient amount of accumulated practice (Newell, 1991).

In the early 1990s, research found that highly experienced and educated experts did not necessarily perform as well as their less skilled counterparts (Camerer and Johnson, 1991; Bedard and Chi, 1992). Bedard and Chi (1992) also questioned the relationship between laboratory studies and real-world expertise and described situations in which novices can not only perform as well as experts but actually surpass the experts.

In response to this criticism, Ericsson and Smith (1991) suggested the redirection of research towards the study of reproducible superior performance in a particular domain, instead of studying the behavior of socially recognized experts. Research during the last three decades demonstrates that the relationship between the amount of accumulated professional experience and actual observed performance is low and may sometimes be negative (Ericsson, 2008).

2.3 Study of expert performance

The study of expert performance starts with the accumulation of a body of reproducible empirical phenomena (Ericsson, 1996). Therefore, methods for reproducing the superior performance under standardized conditions are needed to study the individual differences. If superior performance can be replicated in laboratory-like conditions, the mechanisms underlying superior performance can be investigated by analyzing its structure with experimental methods (Ericsson, 2008).

In sports, the systematic measurement of human performance has a long tradition. In Greek athletic competitions during the Archaic period (700-480 BCE), the fastest runner for 200-meter sprint, 400-meter run, and *distance race* (approximately 5000-meter run), was determined by having all competitors line up at the starting line, allowing all competitors to start at the same time. The evaluation of competitors' performance was based on a pre-established finish line, which allowed the crowd to assess which runner passed the finishing line first. The competitions were conducted in standardized conditions on built, flat tracks instead of on natural terrain (Kyle, 2013).

Since Greek athletic competitions, throughout history, various competitions have been widely held in sports, music, chess, and other domains to find the best performer in a specific domain. The common factor in these domains is that elite individuals reliably outperform less accomplished individuals (Ericsson, 2008).

The scientific study of expert performance strives to identify the mechanisms underlying verified superior performance. In a groundbreaking series of studies, Dutch chess master and psychologist de Groot (1946 / 1978) was able to capture the superior performance of world-class chess players using standardized tasks. de Groot identified critical chess positions from observing games between chess masters and set up a controlled laboratory experiment where chess players were sequentially presented with the associated positions. In this pioneering approach, de Groot (1946 / 1978) also asked chess players to think aloud to study the reasoning of the subjects when they made the best possible next move.

In 1973, new research on expertise emphasized improvements in performance owing to extensive experience in the domain (Simon and Chase, 1973). The focus was not yet on the cognitive processes that mediate the outstanding achievements of experts but on basic memory performance in the laboratory.

In the early 1990s, Ericsson and Smith (1991) characterized the goal of expertise research as being able to "understand and account for what distinguishes outstanding individuals in a domain from less exceptional individuals." The methods used by de Groot formed the basis of the framework called the *expert-performance approach*. In this approach, the real-world performance of experts is scrutinized to identify naturally occurring events that require immediate action and that capture the experts' superior selection or execution of actions in the associated domain.

2.3.1 Expert-performance approach

The expert-performance approach, proposed by Ericsson and Smith (1991), is a systematic framework for the study of expertise. The approach has been used to examine expert performance in a high variety of domains such as sport (Williams and Ward, 2003), music (Ericsson, Krampe, et al., 1993), chess (Charness et al., 1996), education (Plant et al., 2005), and medicine (Ericsson, 2004). The expert-performance approach consists of three main steps:

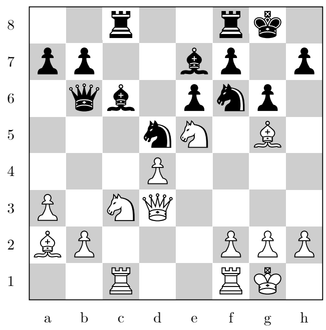
1. **Capturing superior performance** by having expert performers participate in representative tasks and recording the mechanisms that mediate the superior performance,

2. **Identifying the underlying mechanisms** by examining the phenomena associated with a particular type of expertise, and

3. **Examining how expertise developed** by tracing the acquisition of those skills and mechanisms.

First, expert performance in a domain is systematically observed *in situ* to identify standardized tasks that will allow the outstanding real-life performance to be reproduced in the laboratory. The goal is to create a situation that is as simple as possible and yet sufficiently similar to the real-life situation to allow the reproduction of the expertise under laboratory conditions. Figure 2.1 shows three examples of laboratory tasks that capture expert performance. In the first example, de Groot (1946 / 1978) asked chess players to select the best chess move for given positions. In the second example, typists were asked to type as much of the presented text as possible within a minute. In the third example, Gabrielsson (1987) asked pianists to play a Mozart piano sonata multiple times in the same manner.

Figure 2.1: Three examples of standardized laboratory tasks that capture the expert performance of domain experts in chess, typing, and music. Redrawn from "Expertise" by Ericsson and Lehmann, 1999, *Encyclopedia of Creativity*, p. 703.

Domain	**Presented information**	**Task**
Chess	White to move 	Select the best chess move for this position
Typing	OVERVIEW NATURE AND NURTURE OF EXPERTISE The central challenge for any account of expertise is to explain how some individuals attain the highest levels of achievement in a domain and why so few reach that level However, given the continuing struggle in Psychology to explain every day (lower) levels of achievement, it may appear presumptuous to attempt to explain even more advanced levels Consequently, the accounts of expertise have been focusing on the general characteristics of the mechanisms In order to be able to achieve at very high (expert) levels in domains of expertise both nature and nurture are nessary Hence, everyone agrees that experts need to have acquired the necessary domain specific knowledge and skills (nurture) Furthermore, the expert's performance often looks effortless and their most refined and insightful behavior is generated rapidly and naturally rather than the result of prolonged deliberation It would thus appear that experts must excel in general basic characteristics, such as intelligence, memory, speed and flexibility, which have been assumed to be impossinle to train and	Type as much of the presented text as possible within one minute
Music	 Mozart's Piano Sonata in A Major, K. 331	Play the same piece of music twice in same manner

In the second step of expert-performance approach the focus is on discovery of mechanisms underlying the superior performance. In the task analysis, the researcher attempts to identify the most critical aspects of superior performance in the representative task. According to the approach, the full range of methods of analysis in cognitive psychology can be applied to examine the phenomena associated with a particular type of expertise (Ericsson and Smith, 1991). For instance, the differences in cognitive processes between the experts and less-accomplished individuals can be identified by analyzing concurrent verbalizations of thinking before and during the performance (Ericsson, 2006a). Protocol analysis (Ericsson and Simon, 1984) has become a crucial methodological part of understanding experts' cognitive mechanisms that mediate superior performance.

In the third step of expert-performance approach, the focus is on understanding the acquisition of the mechanisms identified in the previous step. The development of the performance and the underlying mediating mechanisms are traced to identify when and how they were acquired. Figure 2.2 shows how the expert-performance approach works backward from the attained expert performance to previous levels of performance to determine when the key mediating mechanisms were first evidenced and whether – and if so, how – they were improved in response to various types of practice (Ericsson, 2020).

Figure 2.2: Illustration of difference between the traditional laboratory research on practice and expert-performance approach. Expert-performance approach has interest in mediating mechanisms that were found to be acquired in response to particular types of practice, whereas traditional laboratory research focuses on the initial acquisition of performance. Redrawn from "Towards a science of the acquisition of expert performance in sports: Clarifying the differences between deliberate practice and other types of practice" by Ericsson, 2020, *In: Journal of Sport Sciences, 38*, p. 160.

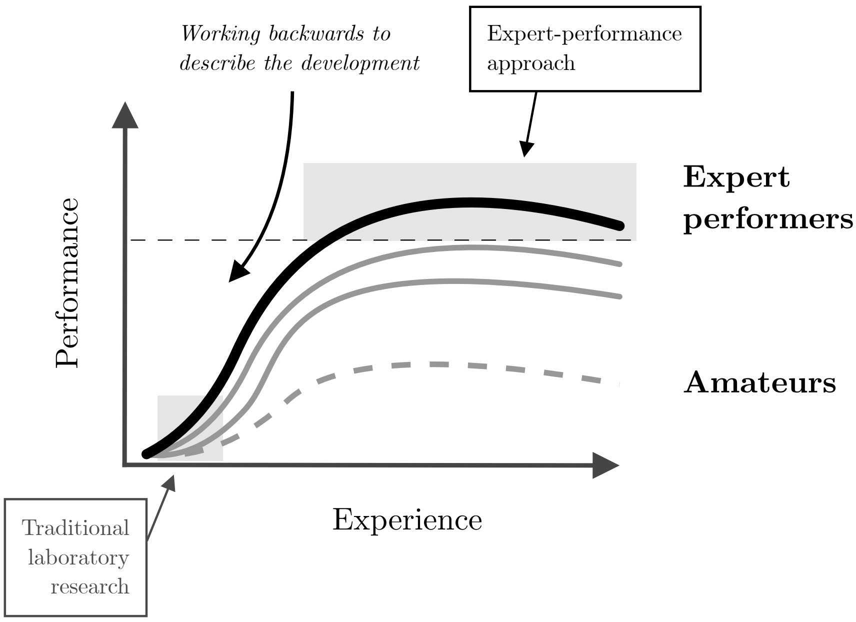


Figure 2.2 also illustrates the fundamental difference between the expert-performance approach and traditional laboratory research (Fitts and Posner, 1967) on practice. In the 1980s, the studies on practice and learning mostly focused on college students learning to perform simple tasks in the laboratory (Schmidt and Bjork, 1992). The tasks were designed to minimize individual differences in participants' prior skills to increase the reliance on essential learning. The goal of these studies was to identify factors that influenced learning after only one or two hours in a generalizable manner across tasks. In contrast, the expert-performance approach involves identifying experts, capturing their superior performance, identifying the underlying mechanisms, and working backward to previous times to understand when and how the performance improved.

2.3.2 Deliberate practice theory

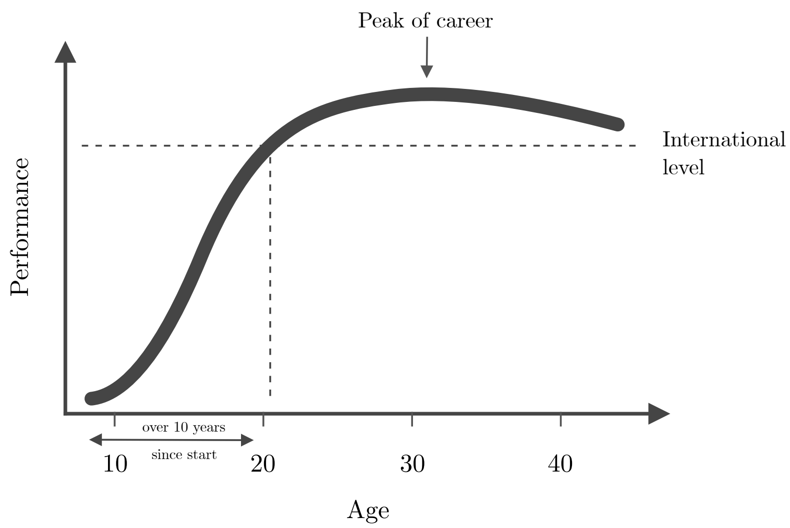
The deliberate practice theory was introduced by Ericsson, Krampe, et al. (1993) as they published the results of their search for the most effective forms of training in music. Since the landmark publication of “The Role of Deliberate Practice in the Acquisition of Expert Performance” by Ericsson, Krampe, and Tesch-Römer (1993), the concept has received much attention and become the *de facto* approach in expertise research. According to Google Scholar, in May 2020, the article had over 10,000 citations.

Deliberate practice was presented as a theory for optimal learning and performance improvement. According to the original definition, deliberate practice is “a regimen of effortful practice designed to optimize improvement,” and significant improvements in performance were observed when individuals were

1. given a task with a well-defined goal,
2. motivated to improve,
3. provided with feedback, and
4. provided with sufficient opportunities for repetition and gradual refinements of their performance.

According to Ericsson, Krampe, and Tesch-Römer (1993), reaching an international top level in established sports, sciences, and arts requires an extended period of deliberate practice. Figure 2.3 shows how all performers, even the most “talented,” need a minimum of approximately 10 years of intense, thoughtful, and focused involvement before being able to be among the top performers. During their development, superior performers often engage in more than 10,000 hours of practice. The accumulated practice frequently accounts for approximately half of the total variance in performance among accomplished individuals, such as full-time music academy students and participants in chess tournaments. By definition, deliberate practice is a “structured activity, often designed by teachers or coaches with the explicit goal of increasing an individual’s current level of performance … Furthermore, deliberate practice involves trying to exceed one’s previous limit, which requires full concentration and effort.” (Ericsson and Lehmann, 1999)

Figure 2.3: Illustration of the increases in expert performance as a function of age. Reaching international level requires typically over 10 years of full-time engagement in high-effort training. During their development, superior performers often engage in more than 10,000 hours of practice. Redrawn from "Expertise" by Ericsson and Lehmann, 1999, *Encyclopedia of Creativity.*



However, the original definition was derived from the domain of music. Music academies have a long tradition in providing students individualized instruction and identifying goals for their practice between meetings with master teachers (Ericsson and Harwell, 2019). Many other domains lack such a teacher-centered determined improvement in areas of development. This has led to misunderstandings and criticism of the theory. By definition, deliberate practice should involve training individualized by a well-qualified teacher, and that criterion is often not fulfilled in other domains (Ericsson and Harwell, 2019). Some recent studies (e.g., Macnamara, Moreau, et al., 2016) have been using the term "deliberate practice" to refer to a wider selection of training activities, adding up all hours of any consistent practice (Ericsson, 2016). That definition significantly differs from the original, where subjects should repeatedly perform similar tasks to improve their performance and receive immediate informative feedback on their performance (Ericsson, Krampe, et al., 1993; Ericsson and Harwell, 2019). To address this conceptual confusion, Ericsson and Pool (2016) assigned distinct names to different types of practice:

- **Deliberate practice**: practice activities meeting all the original criteria,
- **Purposeful practice**: practice activities where participants are engaging in solitary practice to improve particular aspects of performance without regular

guidance by a teacher, and

- **Naïve practice**: work and play activities that are motivated by other factors than the goal of improving a particular aspect of performance.

In summary, the framework offered by the expert-performance approach and deliberate practice concept has been able explain large individual differences in performance in terms of the accumulated consequences of individual differences in sustained activity and deliberate practice (Ericsson, 2008). The common misconception that some uniquely “talented” or “gifted” individuals can reach superior performance in a given domain without much practice is a destructive myth that discourages people from investing the required effort to reach expert levels of performance (Ericsson and Ward, 2007). The acquisition of superior performance takes both time and effort.

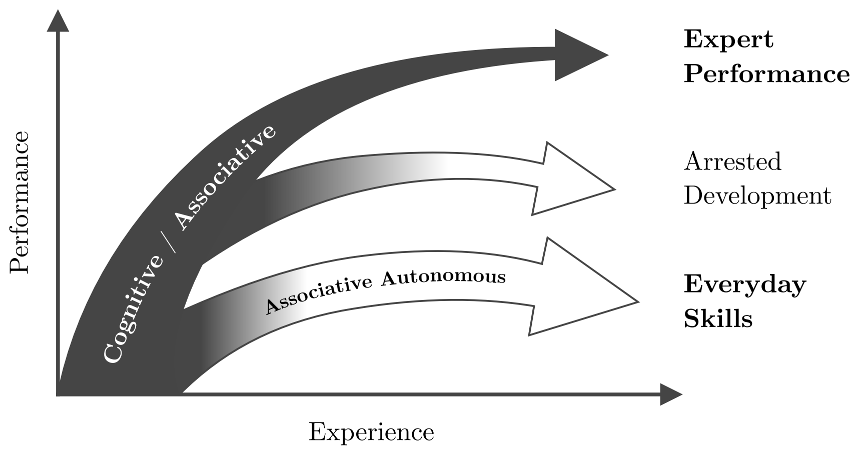
2.4 Experience versus deliberate practice

In an attempt to reconcile the facts that 1) there is no relationship between the amount of experience and measured reproducibly of superior performance in multiple professional domains and 2) there is a need for over 10 years of full-time engagement in high-effort training for reaching high levels of performance, Ericsson, Krampe, et al. (1993) identified the domain-related activities necessary for improving performance and classified them as deliberate practice. Ericsson and Pool (2016) subsequently introduced the term purposeful practice.

As described in the previous section, the exact type of practice is the critical factor. Once a professional reaches an acceptable skill level, more experience does not, by itself, lead to improvement (Ericsson, 2008). Studies have reported that auditors with years of experience were no better at detecting corporate fraud than newly trained novices (Bédard et al., 1993); surgeons were no better at predicting hospital stays after surgery than residents were (Camerer and Johnson, 1991); and physicians’ scores on tests of medical knowledge declined with experience (Choudhry et al., 2005). In many cases, people with extensive training and experience are no better performers than those with minimal experience.

Several studies and reviews (Ericsson, 2006a; Ericsson, 2006b) have suggested a consistent relationship between the quantity and quality of solitary activities, meeting the criteria of deliberate practice and performance in a wide range of domains of expertise. Figure 2.4 illustrates how everyday performance shows an initial increase as the subjects expand their level of effort to reach an acceptable level. As the subjects adapt to the performance demands, their actions can be carried out with reduced attentional control. At the same time, as the subjects' behavior is automatized, the current behavioral repertoire becomes fixated, and the subjects lose their conscious control over intentionally modifying and changing it. Under these circumstances, further experience is not associated with any improvement or learning. Consequently, the correlation between the amount of experience and performance is low for this type of automated everyday activity. In direct contrast, expert performance continues to improve as a function of more experience and deliberate practice (as illustrated in Figure 2.4). The critical challenge for aspiring expert performers is to avoid the arrested development associated with automaticity and to acquire cognitive skills to support continued learning and improvement (Ericsson, 1998).

Figure 2.4: Illustration of the qualitative difference between the course of improvement of expert performance and of everyday activities. Redrawn from "Expertise" by Ericsson and Lehmann, 1999, In: *Encyclopedia of Creativity*.



The popular book "Outliers" (Gladwell, 2008) introduced the "10,000-hour rule", using findings by Ericsson, Krampe, and Tesch-Römer (1993) as primary empirical evidence of its existence. Because the most accomplished musicians had over 10,000 hours of practice by age 20 (Ericsson, Krampe, et al., 1993), Gladwell concluded that "ten thousand hours is the magic number of greatness" and considered it to be the key to success in any field, as long as an individual keep practicing twenty hours a week for ten years. The evident drawback of the rule is that it focuses on the amount of time spent practicing and not on the quality of that practice (Ericsson and Harwell, 2019; Ericsson, 2020).

To summarize the above, the prevalent focus seems to be on how long one has been practicing although the critical factor is the type of practice (Ericsson and Harwell, 2019). Over the ages, children have been told that if they focus on working hard, they will be fine. According to decades of research on expertise, they will be fine but most likely never exceptionally good. Rather than hard work and the exact number of hours, the factor that best explains exceptional performance is the quality of practice. When this condition is fulfilled, the quantity also begins to be of significance.

2.5 Expertise in computer use

There is no prior research available on skills in system administration. The limited research on system administration has focused mostly on work practices (Barrett et al., 2004), education (Kuncicky and Wynn, 1998), preferred user interfaces (Takayama and Kandogan, 2006; Voronkov et al., 2019), and research methods (Anderson and Patterson, 2002). However, the literature on skills in computer use and the expertise studies in similar technical problem-solving domains, such as programming, provide an opportunity to better understand the characteristics of expertise and its development in technical professions.

2.5.1 Acquisition of skills in information and communication technology

In certain aspects, system administration is markedly different from the other domains investigated in expertise research, such as sport and music. In system administration and other information and communication technology (ICT) professions, the role of skill is more practical and instrumental. Research indicates that people are not interested in understanding computers *per se* but in understanding how computers can help them accomplish their primary goals. Facer et al. (2001) investigated why young people might value and acquire computer expertise and found that they are motivated to achieve goals that are practical by nature and align with their future plans concerning ICT in their further education and profession. The development of ICT expertise of students was found to be based on strong internal motivation, intensive use of ICT, and informal learning at home. Similarly, a study of experts by Ericsson and Lehmann (1996) showed that they achieved high-level results owing to a combination of strong motivation and concentration. Ilomäki and Rantanen (2007) studied the development of computer skills in lower secondary school students; they also found that high-level skills were typically informally acquired at home rather than through formal education. The acquired, relatively advanced capability of young students was considered a kind of expertise, and it is often valued in their social environment. The acquisition of ICT skills seems to be associated with gender because many studies have reported measurable differences in ICT skills between genders: boys have better ICT skills, they use ICT more in their leisure time, their attitudes toward ICT are more positive, they use ICT more for playing and recreational purposes, they are more interested in hardware, and they take on more independent challenges for learning ICT than girls do (Ilomäki and Rantanen, 2007; Kaarakainen et al., 2018).

Unlike in many competitive domains, such as sports and music, the skills in system administration and other practical domains are not typically practiced to outperform others. The conditions available for learning ICT skills are considerably different from those typically available to individually trained individuals, such as music academy students. However, the deliberate practice theory emphasizes the goals set and informative feedback provided by external agents such as teachers and coaches. In technical domains, individuals often do not have teachers or coaches to prepare exercises and provide feedback; the feedback is received from the system itself and through self-regulated learning. Research indicates that the same deliberate practice mechanisms apply for self-set goals (Ericsson and Lehmann, 1996), and that the improvements in performance can be attributed to self-directed practice activities that fulfill the criteria for deliberate practice (Ericsson, 2004). Because the ICT skills are reported to be acquired informally and not through formal education, understanding the process of self-regulated learning is necessary.

Self-regulated learning is a cyclical process wherein the students plan for a task and monitors their own performance; it then affects the outcome – the cycle then repeats as the student uses the effect on the outcome to modify and prepare for the next task (Zimmerman, 2002). Multiple studies on self-regulated learning (for review see Puustinen and Pulkkinen, 2001) in ICT report that stable improvements can be achieved (Brand-Gruwel et al., 2005). Self-regulated learning involves more than detailed knowledge of a skill; it involves self-awareness, self-motivation, and behavioral skill to appropriately implement that knowledge. According to Zimmerman (2002), the experts can spend multiple hours each day in study and practice and find these activities highly motivating. They vary their methods of study and practice in order to discover new strategies for self-improvement.

2.5.2 Research on programming skill

From the perspective of expertise research, computer programming as a complex problem-solving activity has many similarities with system administration. From a psychological perspective, many software design and system administration tasks can be described as ill-defined problems (Simon, 1973), which implies that problem specifications are incomplete and must be decided on during the process. Therefore, in contrast to the tasks in distinct domains, ill-defined problems have no single correct solution. This makes the research and quantification of the expertise more complex.

In programming, large differences in skill levels have been observed among programmers (Mayer, 1997; Carroll, 1997). Extensive research on expertise on programming started in the early 1980s (Sonnentag et al., 2006). Jeffries, Turner, Polson, and Atwood (1981) and Adelson (1984) published studies on how experts differ from novices in programming skills. These studies stimulated subsequent research and were often cited in more general publications on expertise (Ericsson and Smith, 1991). In the domain of programming, as in many other domains, researchers have relied on traditional conceptualization and operationalization of expertise (Sonnentag et al., 2006). Sonnentag (2001) reported that 84% of all quasi-experimental studies on expertise in software development published between 1981 and 1997 used an operationalization of expertise that was based on months and years of experience. In the domain of programming, as in many other domains, researchers have relied on traditional conceptualization and operationalization of expertise (e.g. Jeffries et al., 1981). Most studies on programming have reported that more advanced programmers outperform relatively inexperienced programmers with respect to performance quality and solution time (Sonnentag et al., 2006). Sonnentag et al. (2006) has reviewed previous research on programming skills and provided an overview of the results. Programming skills are divided into five areas: 1) requirements analysis and design tasks; 2) programming and program comprehension; 3) testing and debugging; 4) knowledge representation and recall; and 5) communication and cooperation. High performers tended to spend less time on problem comprehension, pursue abstract programming goals, use a cross-referencing strategy (Pennington, 1987), search for problems, show broader and more detailed knowledge base, and spend more time on communication and cooperation.

2.6 Applicability of research methods to system administration

Because skills in system administration have not been previously studied, it is essential to critically analyze the selected methods and their potential and applicability to answer the research questions. The selected framework – expert-performance approach – has been used to examine skills in several domains including sports (Williams and Ward, 2003), music (Ericsson, Krampe, et al., 1993), games (Charness et al., 1996), medicine (Ericsson, 2004), typing (Keith and Ericsson, 2007), education (Plant et al., 2005), and using smartphones (Oulasvirta et al., 2011). The three-stage *de facto* approach consists of 1) identifying representative tasks, 2) having expert performers participate in the tasks and recording the mechanisms that mediate superior performance, and 3) tracing the acquisition of skills and mechanisms (Ericsson and Smith, 1991). Each of the three stages involves considerations and critical issues in examining skills in system administration.

First, regarding skills, system administration is an extremely complex and wide-ranging domain. Compared with competitive domains, in which precise rules define the ideal and measurable outcome and the ones who perform best according to the established rules can be declared as experts, system administration as the upkeep, configuration, and reliable operation of computer systems is considerably more challenging to condense into representative tasks. Second, measurement of performance in solving ill-defined problems – which have no single correct solution – is clearly more complicated than measuring that in typing or other everyday task, where the participant's performance can be assessed unambiguously. Third, examining the expertise development of system administrators is more challenging than that in athletes or musicians, who typically have regular competitions, level tests, and training diaries to retrospectively examine the amount of training and level of results. Such mechanisms and practices are entirely lacking in the domain of system administration. Therefore, expressing the skill level of system administrators as a function of time and being able to identify when and how these skills developed does not seem entirely feasible.

Moreover, the original deliberate practice theory (Ericsson, Krampe, et al., 1993) is based on findings in pianists and violinists in music academies. Identifying deliberate practice, which is often deemed the critical factor for optimal learning and performance improvement, has been challenging in other domains than music (Macnamara, Moreau, et al., 2016). The original definition of deliberate practice requires such teacher involvement and individualized training activities that fulfilling all the criteria has not been achievable in other domains, which has led to debate (Hambrick, Altmann, et al., 2014; Ericsson, 2016; Macnamara, Moreau, et al., 2016) and a restructuring of the theory (Ericsson and Pool, 2016). It is assumed that practice in an ICT profession such as system administration does not fulfill the stringent criteria that are not fulfilled in the training of professional athletes. Furthermore, in many domains in which deliberate practice has been studied, most of the variance in performance has been explained by factors other than deliberate practice (Macnamara, Hambrick, and Oswald, 2014), including initial age (Gobet and Campitelli, 2007), working memory (Meinz and Hambrick, 2010) capacity, and genetics (Hambrick and Tucker-Drob, 2015).

These concerns can be taken into account and addressed in the design of the experiment. There exists a broad consensus on the essential skills of a system administrator, and the performance can be captured in standardized conditions. By carefully designing tasks and their goals, success can be measured by assessing the system condition. Furthermore, the use of process-tracing methods, protocol analysis, and retrospective interview methods has allowed researchers to capture the performance and its development in numerous complex domains. System administration is a complex, engaging, and relevant area of research. The research on skills in system administration may not be easy and effortless, but generalization of findings would have comprehensive benefits. Therefore, a carefully designed and implemented research method is presented in the next chapter.

3 Method

The study included two parts: 1) measurement of task performance while thinking aloud; and 2) interviews focusing on the six subcomponents (see Table 3.11) of users' practice. The latter part of the study was conducted within two months of the first part.

3.1 Participants

20 professional system administrators were recruited for the experiment. Most of the participants were recruited by contacting the information technology (IT) department heads of southern Finland's universities and companies. The department heads were asked to nominate system administrator candidates who were the top performers in their organizations. Other recruitment methods were calls for participation across multiple digital channels such as sysadmin mailing lists, online forums, Helsinki Institute for Information Technology's (HIIT) website, Usenet newsgroups, and relevant Internet Relay Chat (IRC) channels, and peer referrals (where the participants were asked to name the most skilled system administrators they knew).

To recruit the most suitable and eligible participants, all the candidates were asked to provide the following information:

- total years worked in the IT industry;
- total years in a system administrator role;
- whether they were currently working as a system administrator;
- current location;
- current duties as a system administrator;
- a description of the system they currently worked with;
- educational details (what, where and when);

- other background information (courses, certifications, computer-related hobbies); and
- availability during the experiment.

To be selected, the candidate had to be available, willing to participate, and meet most of the selection criteria. The primary focus was on versatile, educated, and experienced candidates. Such individuals were typically administrating a complex IT system (as opposed to simply performing helpdesk tasks) and were the trusted ones who were approached by their organizations when a complex IT problem arose. Participants were sought from both companies and universities.

Because the study focused on a system administration domain, it was reasonable to only include participants whose characteristics were sufficient to investigate the research problem (Purchase, 2012). These characteristics included full-time employment as a system administrator and several years of work experience in the IT industry. The number of participants required for a study depends on the design of the study; a within-subjects design needs fewer participants than a between-subjects design in order to be statistically significant. According to Hornbæk et al. (2013), HCI studies typically use 20 participants – having too few participants is simply not powerful enough to detect the effects.

The mean ages of the participants was 34.9 years ($M = 37$, $SD = 6.7$), with an age range of 22 to 42 years. Two of the subjects were female, and eighteen were male. Fifteen subjects had a university education, two subjects had a vocational education, and three subjects had no formal IT-related education at all. According to the demographic information collected, the participant population was a very representative sample when measured against the annual surveys conducted by the USENIX Special Interest Group for Sysadmins (2011).

On average, participants had 13.5 years ($M = 14.5$, $SD = 6$) of work experience in the IT industry and 11.2 years ($M = 13$, $SD = 5.8$) of professional system administration experience. All the participants were currently working as professional system administrators. Seven of the participants were currently working in universities (including University of Helsinki, Aalto University, University of Tampere, and the Helsinki Institute of Information Technology), six at large companies, and seven in small and medium-sized enterprises.

Participants received a reward of 100 euros each for their participation.

3.2 Experiment

3.2.1 Design

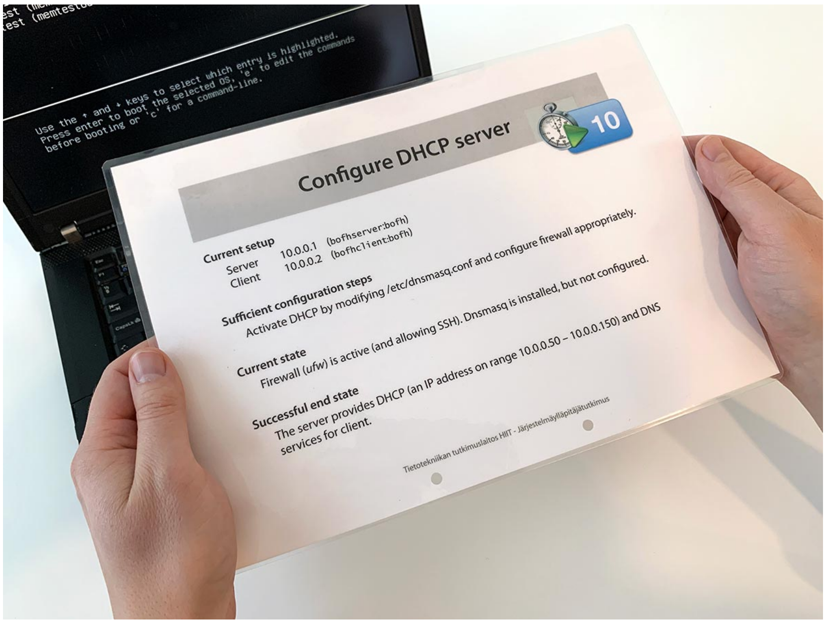
This study used a within-subjects design — a type of experimental design in which all participants are exposed to every treatment or condition. The independent variables were 1) task and 2) indicators of users' expertise and experience collected in the interviews. Dependent variables were indicators of task performance, events in verbal protocols, and actions taken during the performance.

3.2.2 Tasks

To study system administrators' performance, nine representative tasks were designed to measure their skills. The tasks were divided into three blocks, and the order of the latest two blocks was randomized. Each task had a threshold time of either five, 10, or 15 minutes. The combined maximum execution time for all the tasks was 90 minutes. To generate a representative, realistic, modern, and suitably challenging set of tasks, while avoiding ceiling and floor effect, a prior study from HIIT was utilized. Based on system administrator interviews, 25 typical tasks were identified to measure the system administration skills. Based on interviews and pilot tests with professional system administrators, nine of the 25 tasks were selected for the study. In the pilot tests, a large number of tasks were evaluated, and system administrators were consulted on whether the tasks were relevant, challenging enough, and feasible to implement within the experiment setup.

Tasks were introduced to subjects using task cards (Figure 3.1), which included information about the current setup, sufficient configuration steps, current system state, and definition of successful end state. All the task cards are included in Appendix A.1.

Figure 3.1: A4-sized task card.



Block 1: Networking fundamentals

The first block consisted of three computer networking tasks. Maintaining and administering computer networks — in other words, configuring firewalls, Dynamic Host Configuration Protocol (DHCP) servers, and Domain Name System (DNS) services — are the most traditional and fundamental tasks of the system administrator.

The total available execution time for the first block was 30 minutes. The tasks of the first block were a continuation of one another, and the next task continued from the previous task's successful end state.

The goal in Block 1 was to allow Secure Shell (SSH) traffic, enable the firewall, enable the DHCP server to assign IP address to the client dynamically, and to forward client's traffic through the server. The tasks of Block 1 are described in Table 3.1, Table 3.2, and Table 3.3.

Title	**Enable firewall and allow SSH access**
Task	1/9
Maximum time	5 minutes
Initial state	Both client and server are up and running. Server's firewall (ufw) is installed, but inactive.
Successful end state	Firewall is activated and SSH access is allowed.
Steps required	Opening connection to server (`ssh bofhserver@10.0.0.1`), activating firewall (`sudo ufw allow ssh`) and allowing SSH access to server (`sudo ufw enable`).

Table 3.1: Task 1 (Networking fundamentals)

Title	**Configure DHCP server**
Task	2/9
Maximum time	10 minutes
Initial state	Firewall (ufw) is active (and allowing SSH). Dnsmasq is installed, but not configured.
Successful end state	The server provides DHCP (an IP address on range 10.0.0.50 – 10.0.0.150) and DNS services for client.
Steps required	Opening connection to server, setting DHCP range by modifying `/etc/dnsmasq.conf`, restarting service, and adding firewall rules for DHCP (`sudo ufw allow bootps`) and DNS (`sudo ufw allow domain`).

Table 3.2: Task 2 (Networking fundamentals)

Title	**Enable Internet sharing**
Task	3/9
Maximum time	15 minutes
Initial state	Dnsmasq is installed and partially configured. Server provides DHCP (an IP address on range 10.0.0.50 – 10.0.0.150) and DNS services for client. Server has internet connection through LAN. Server and client are properly connected. Client has no other network connection.
Successful end state	Client's traffic is forwarded through server, enabling access to the internet.
Steps required	Opening connection to server, activating forwarding (modifying `/etc/ufw/sysctl.conf`), changing default forward policy (modifying `/etc/default/ufw`), and adding NAT rule (utilizing either ufw's `before.rules` or `iptables`).

Table 3.3: Task 3 (Networking fundamentals)

Block 2: Software management

The second block consisted of three software and package management tasks. The ability to understand, modify, and debug programs, as well as proficiency with the installation and administration of application software, are essential technical skill requirements for the well-rounded system administrator (Kuncicky and Wynn, 1998). In the second block, each subject first wrote and compiled a simple program, turned it into a deb package, and added the package to the software management system.

The total available execution time for the second block was 25 minutes. The tasks of the block were a continuation of each other, and the next task continued from the previous task's successful end state.

The tasks of Block 2 are described in Table 3.4, Table 3.5, and Table 3.6.

Title	**Write Hello World program and compile it**
Task	4/9
Maximum time	5 minutes
Initial state	Server is up and running and has a ready-formatted working directory.
Successful end state	The Hello World program is compiled and it can be run from command line.
Steps required	Writing a small Hello World program using C language, creating a `Makefile` file and compiling (`gcc` and `make`) the program.

Table 3.4: Task 4 (Software management)

Title	**Create a software package in deb format**
Task	5/9
Maximum time	10 minutes
Initial state	Server is up and running and has a ready-formatted working directory of a small executable (`~/helloworld/usr/bin/helloworld`) that needs to be packed in a deb binary package.
Successful end state	A deb package file is containing the mentioned executable from working directory. If deployed with deb package manager to some other machine, the executable in installed the same way with binary executables in that system.
Steps required	Opening connection to server, creating an appropriate `~/helloworld/DEBIAN/control` file, and creating the deb binary package (`dpkg-deb -build`...).

Table 3.5: Task 5 (Software management)

Title	**Add software package to the software management system**
Task	6/9
Maximum time	10 minutes
Initial state	The deb package file has been copied to the "repository". The web server is running (Apache is installed and firewall is configured), acting as software repository appropriately. The package manager in the client has been configured (`/etc/apt/sources.list`) to use this repository.
Successful end state	The `Packages.gz` file has been created. The Helloworld package is available from our self-managed and unofficial software repository. Helloworld package has been installed on the client using the current repository.
Steps required	Opening connection to server, creating index file for binary package (`dpkg-scanpackages`...), and installing Helloworld package to client (`apt-cache search` and `apt-get install`).

Table 3.6: Task 6 (Software management)

Block 3: Automation and scripting

Proficiency in authentication schemes, configuring network file systems, and setting up data backup automation are some of the more challenging tasks for a system administrator. The ability to program in an administrative scripting language is a typical requirement for system administrator.

The total available execution time for the third block was 35 minutes.

The tasks of Block 3 are described in Table 3.7, Table 3.8, and Table 3.9.

Title	**Write a shell script to change file access permissions**
Task	7/9
Maximum time	5 minutes
Initial state	System is up and running.
Successful end state	Using your script, group can be given read/write access to all files of specific user.
Steps required	Write a small shell script to recursively give group read/write permissions to all specific user's files.

Table 3.7: Task 7 (Automation and scripting)

Title	**Setup the client to mount home directories from the server**
Task	8/9
Maximum time	15 minutes
Initial state	The client is up and running, and has default user authentication. A PAM module is installed. Server has a Samba SMB/CIFS fileserver running, sharing user home directories. Client and server have a network connection.
Successful end state	On client, home directories are mounted from the Samba service (using single password) automatically when user logs in.
Steps required	Testing SMB share (`smbclient`) and adding volume definition to `pam_mount.conf.xml` to mount home directories from the server.

Table 3.8: Task 8 (Automation and scripting)

Title	**Set up a backup routine for a laptop**
Task	9/9
Maximum time	15 minutes
Initial state	The client has users' files in `/home` directory. Client and server have a working network connection. The data to be backed up can be stored on the server. Backup utility (`rsnapshot`) is installed and configured.
Successful end state	The client's `/home` directory is backed up automatically every time a network connection to the server is established, but not more often than once a day.
Steps required	Creating backup script that does two checks before executing backup command (`rsnapshot daily`): checking connection to server (e.g. `ping` exit value 0), and checking that the log file (`/var/log/run-my-backup.log`) is older than a day utilizing `find -mtime`, exit value, piping or similar approach. Finally, if `rsnapshot` backup exits successfully (exit value 0), touching or modifying `/var/log/run-my-backup.log`, and soft-linking the script to `/etc/network/ip-up.d/` directory.

Table 3.9: Task 9 (Automation and scripting)

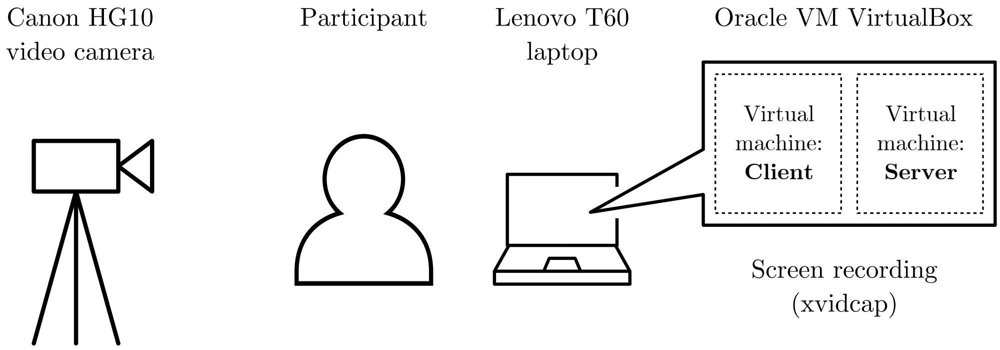
3.2.3 Setup and apparatus

The experiments took place in HIIT's Espoo office, the University of Helsinki Kumpula campus, the University of Tampere, and in several offices in Helsinki area.

During the experiment, each subject used a Lenovo Thinkpad T60 laptop (Intel Core 2 Duo T7200 2 GHz processor and 512 megabytes of RAM) running Ubuntu Linux operating system, Oracle VM VirtualBox with two virtual machines, and screen recording software (xvidcap). The laptop had wireless 3G Internet connectivity. The whole experiment session was videotaped with a Canon HG10 digital video camera (Figure 3.2).

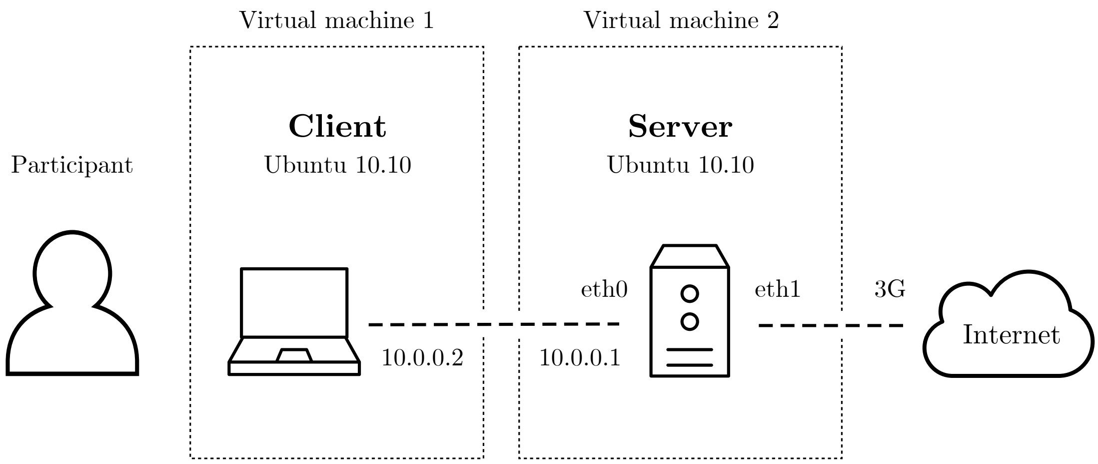
Subjects were allowed to use their own devices for information retrieval and for anything else they desired during the experiment.

Figure 3.2: Experiment setup



The experiment tasks were conducted in a virtualized environment (Figure 3.3) where the participants performed the given system administration tasks (as described in Section 3.2.2) The virtual machine-based approach allowed us to restore the preconfigured system states (snapshots) to provide immutable and appropriate initial client and server states for each task regardless of the participant's activities in previous tasks, made the setup lightweight and portable, and made the recording of the screens and executed commands easy. Both virtual machines ran standard Ubuntu operating systems, which is a popular Debian-based Linux distribution.

Figure 3.3: Experiment system architecture

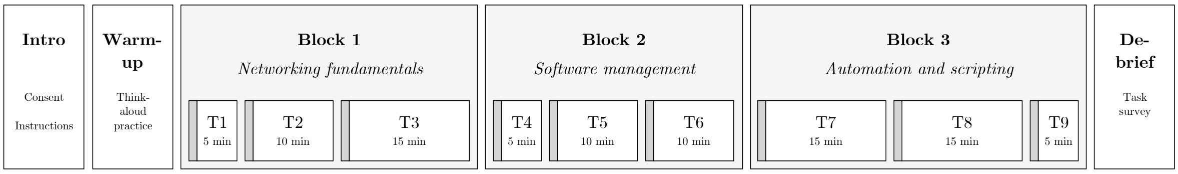


3.2.4 Procedure

The total duration of a single session, including the instructions, warm-up tasks, anticipations, task performance, and retrospective reports, was approximately two hours. The procedure is visualized in Figure 3.4.

Figure 3.4: Experiment procedure

Part 1: Measurement of task performance (120 minutes)



Part 2: Retrospective interviews (30-60 minutes)

During the introductory phase, the subjects were given the participant information sheet (Appendix A.2), and they filled out the consent form (Appendix A.3).

Participants were given the standard instruction to "think aloud" (Ericsson, 2002; Ericsson, 2006a), along with a few warm-up tasks that would familiarize them with thinking aloud while completing simple tasks (e.g., "What letter comes immediately after "A" in the alphabet?" and "What is the fourth letter after 'N'?").

The participants were instructed to act just as if they were alone and speaking to themselves. They were told not to try to explain what they were thinking nor plan what to say, but simply to verbally express their thoughts. Participants were invited to ask questions about the procedure and given further clarification when requested.

Each task had three phases:

1. **Description of the task** on a task card: The participant could ask clarifying questions.

2. **Anticipation**: The participant elicited a prospective solution and described the planned solution path. The participant was asked the predicted probability of success of the task on a scale of 0% to 100%.

3. **Task performance**: A task-specific time (five, 10, or 15 minutes) was given for the achievement of the solution. Any participant was still attempting to complete a task after the maximum time elapsed was asked to stop, and the task was marked as a failure.

The tasks were videotaped and the laptop screen was recorded.

At the end of the experiment session, participants filled the post-test survey (Appendix A.4), where they expressed their perceived representativeness of the tasks.

3.3 Protocol analysis

Protocol analysis is a precise methodology for "eliciting verbal reports of thought sequences as a valid source of data on thinking" (Ericsson and Simon, 1984). Several iterations of data analysis were required to develop a coding manual for behavioral and cognitive events. The final categories used were inspired partially by previous work, (most notably by the predictions of the deliberate practice theory) and in part by what could be encoded reliably from the data with reasonable time investment.

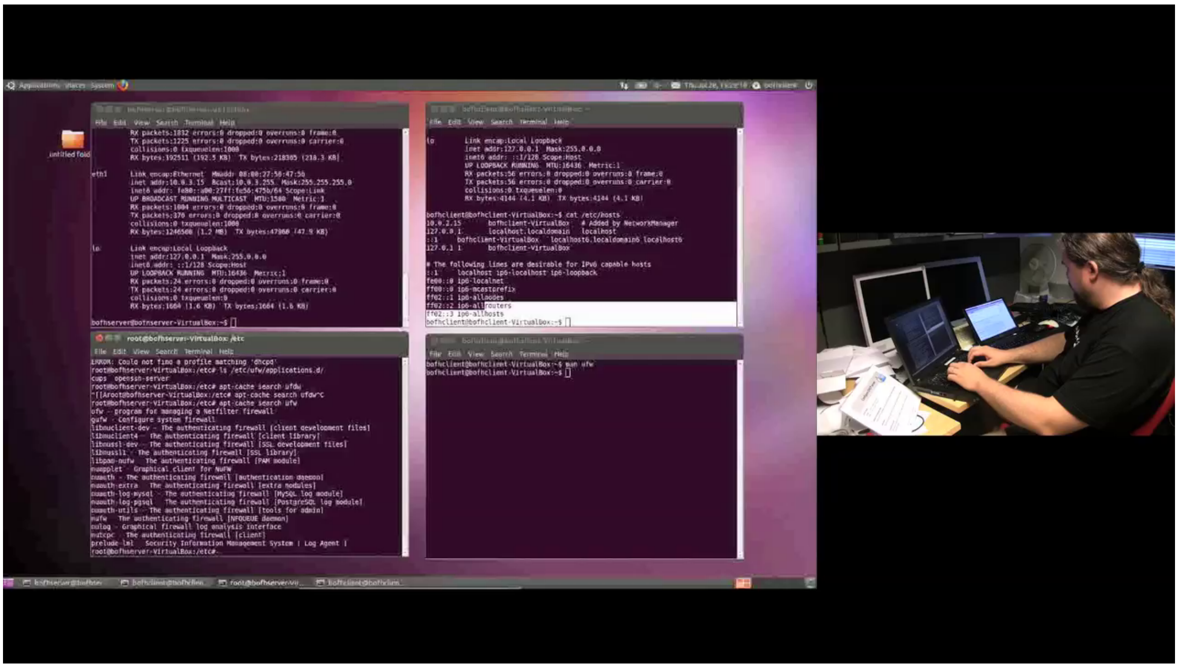
The performance-related categories (Table 3.10) elicited in video and protocol analysis were (1) overall time; (2) completed steps and sub-goals; (3) directly useful actions; (4) harmful actions; (5) information search: system; and (6) information search: external. The thinking-related categories used in video and protocol analysis were (7) planning, (8) confidence, and (9) unconfidence.

Category	Description
Behavioral	
Reached sub-goal	Subject reaches one of predefined sub-goals
Directly useful action	One of predefined *useful actions* occurs
Useless action	Subject does something that changes system state, but does not help anyhow on task completion
Harmful action	Subject does something that needs to be undone to complete the task
Information search: system	Reading documentation (man pages)
Information search: external	Using Internet search.
Testing and debugging	Constructive testing of a solution
Cognitive	
Planning and anticipation	Expressing planned actions to solve the task
Confidence	Expressing confidence
Unconfidence	Expressing uncertainty

Table 3.10: Categories for video data and verbal protocols.

The coding manual, showing examples, is available in Appendix B. The protocol analysis was carried out from the edited video recording (Figure 3.5).

Figure 3.5: Analyzed video material including screen capture, video, and audio.



3.4 Retrospective interviews

The interview method was inspired by the detailed retrospective interview procedure introduced by Côté et al. (2005). The interview's primary focus was on the six sub-components of users' practice (described in Table 3.11).

Quality	Characterization
1. Motivation	Self-motivated; "To be the best in the field"; The explicit goal is to improve performance
2. Concentration	Full concentration and focus on the study activities
3. Design of practice	In the beginning, practice methods designed by an expert/teacher, learning methods invented in the progress of a career, and the individual's weaknesses systematically explored and focused upon
4. Feedback	Informative feedback on results of performance, acquired or given by a teacher
5. Regularity	Habitual practice at regularly scheduled times
6. Emotions	Not as inherently enjoyable as competing activities

Table 3.11: Qualities of deliberate practice (Ericsson, Krampe, et al., 1993; Ericsson, 2004; Ericsson, 2006b)

The main themes in the interview were the participants' backgrounds, skill acquisition routines, the characteristics of the currently administered system, and the level of deliberate practice. The structure of the interview, the questions, and the data types of the answers were as follows:

Section one: **Background**	
Role and experience	
Official title in organization	(String)
Tasks and responsibilities (list all)	(String)
Years working with the current title in the current company	(Integer)
Total years in the current profession	(Integer)
Total years in sysadmin role	(Integer)
Total years in IT industry	(Integer)
Professional experience	
Employer	(String)
Years	(Integer)
Duties	(String)
Education	
Academic degree	(String)
Year	(Year)
Certifications	
Course or certification	(String)
Year	(Year)

Before the skill acquisition section, two new notions were introduced to the subject: *Utility* (perceived value for success in work tasks) and *effort* (perceived investment of effort)

Section two: **Skill acquisition**

Computer related activities that you were involved before a professional career and that you kept doing for an hour or more per week or that you engaged in for more than a total of 50 hours:

	Hours/week (integer)	Years (integer)	Utility (1-5)	Effort (1-5)
Programming				
Gaming				
BBS				
Tech forums				
Hardware tweaking				
Personal system administration				
Other (what)				

Computer related activities that you've been involved during professional career and that you've been doing for an hour or more per week or that you engaged in for more than a total of 50 hours:

	Hours/week (integer)	Years (integer)	Utility (1-5)	Effort (1-5)
Programming				
Gaming				
Tech forums				
Hardware tweaking				
Personal system administration				
Other (what)				

How much of what you do know about the system you currently supervise is due to...

Hands-on experience (with this system and other systems)?	(0-100%)
Formal training programs (e.g., Microsoft, AWS, Sun)?	(0-100%)
Reading and research on your own (e.g., books, internet)?	(0-100%)
Relevant formal education (e.g., university)?	(0-100%)
Working with and learning from others?	(0-100%)
Something not listed above?	(0-100%)

Section three: **Current system**

Generalizability of findings: As far as you can tell, the system you currently supervise is...

a) Unlike any other system in the world	(yes/no)
b) For the most part, unusual	(yes/no)
c) Equal parts unusual and generic	(yes/no)
d) For the most part, generic	(yes/no)
e) Completely standard	(yes/no)

Is the current system...

a) Mostly self-built	(yes/no)
b) Partly self-built and partly inherited	(yes/no)
c) Mostly inherited	(yes/no)

Working environment

How many co-workers you have working on [mostly] same tasks as you?	(Integer)
What parts/products/subsystems do you supervise?	(String)

Section four: **Deliberate practice**

Recent experience

What was the last new product/part-of-the-system you studied?	(String)
When was that?	(Date)
How much time did you spend?	(Hours)
How did you assess the mastery level?	(String)

Solitary time studying

When adopting a new subsystem, how many weekly hours you use for studying it?	(Integer)
How much time you use for experimenting how it works?	(yes/no)
Do you have a specific test environment for such experimenting? (describe)	(yes/no)

When you have to do something to this system or fix something on this system, and you DON'T already know how to do it or fix it, which of the following actions do you take and what percentage of the time do you take them?

a) Consult with people you know who have experience	(0-100%)
b) Contact manufacturer support	(0-100%)
c) Contact third party support	(0-100%)
d) Do research via books or technical literature	(0-100%)
e) Do research via the web newsgroups	(0-100%)
f) Experiment to try and see what works	(0-100%)
g) Use diagnostic tools	(0-100%)
h) Take an action that is not listed above.	(0-100%)

Systematically testing skills

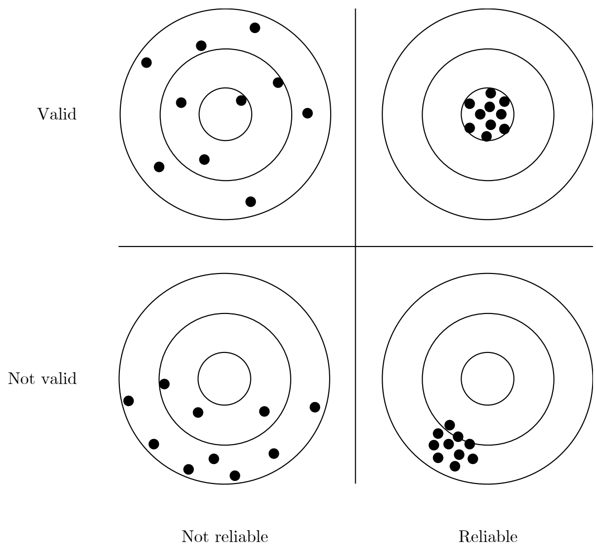
Are you systematically testing your skills, e.g. ”can I do this” (describe how)	(String)

The tape-recorded interviews were tabulated directly and frequencies for each subject were counted.

3.5 Validity and reliability

Validity and reliability are essential in analyzing the appropriateness, meaningfulness, and usefulness of an experiment and its results. Validity is described as the degree to which an experiment measures what it intends to measure. The validity of the design of experimental research is a fundamental part of the scientific method, and also a concern of research ethics — without a valid design, valid scientific conclusions cannot be drawn. Reliability is the overall consistency of a measure. A measure is said to have a high reliability if it produces similar results under consistent conditions. Figure 3.6 illustrates the relationship between validity and reliability.

Figure 3.6: Illustration of the relationship between validity and reliability.



According to Shadish et al. (2002), there are four types of validity: *statistical conclusion validity*, *internal validity*, *construct validity*, and *external validity*. Shadish et al. (2002) have also listed factors that may jeopardize different types of validity and which demanding the attention of researchers.

Statistical conclusion validity is defined as the validity of interferences about the correlation (covariation) between treatment and outcome (Shadish et al., 2002). Typical threats to statistical conclusion validity are fishing (mining the data to find something significant), low statistical power, homogeneity of participants, and unreliability of measures (that can result in overestimating or underestimating the size of the relationship between variables). In this thesis, a relatively large and heterogeneous group of participants was recruited for the experiment, and the measurements were performed reliably to ensure statistical conclusion validity.

Internal validity is defined as the validity of inferences about whether observed covariation between A (the presumed treatment) and B (the presumed outcome) reflects a causal relationship from A to B as those variables are manipulated or measured (Shadish et al., 2002). An artificial laboratory setting typically ensures higher internal validity because external influences can be minimized. Examples of threats to internal validity are selection, history, maturation (e.g., the passage of time influences the dependent variable), and testing (e.g., participants feel the need to be consistent in their behavior in the pre-test and post-test). In this thesis, the experimental part took place over several months and consisted of two parts; therefore both maturation and testing are relevant threats to the internal validity.

Construct validity is defined as the validity of inferences about the higher-order constructs that represent sampling particulars (Shadish et al., 2002). Typical threats to construct validity are inadequate explication of constructs, mono-operation bias, mono-method bias, and experimenter expectancies. In this experiment, the subjects had the opportunity to ask questions during the anticipation phase, and therefore experimenter expectancies posed a threat to construct validity.

External validity is defined as the validity of inferences about whether the cause-effect relationship holds across variations in individuals, settings, treatment variables, and measurement variables (Shadish et al., 2002). In other words, external validity is the extent to which the results of a study can be generalized to and across different situations, people, environments, and technologies (Gergle and Tan, 2014). Threats to external validity include the interaction of treatment and selection, setting, and history. For example, individuals who actively offer to participate in the experiment may differ from the population for which the results are to be generalized. In the recruitment phase of this study, the subjects were approached personally based on referrals. Therefore, even individuals who might not have otherwise volunteered were recruited as participants.

Reliability refers to the extent to which a scale produces consistent results if the measurements are repeated several times. In practice, testing measures are never entirely consistent. However, in this thesis, the time was taken to follow the procedure carefully during execution of the trials with the aim of increasing the reliability of the study. The biggest threat to reliability was the potentially unreliable interpretation of behavioral and cognitive events in the protocol and video analysis. To ensure reliability, the coding manual was developed iteratively to include only observable, unambiguous, definable, and recognizable variables.

4 Results

The analyzed data consisted of a total of 180 unique pre-task anticipations and task performances by 20 participants (with a total duration of 39 hours and 41 minutes), post-trial task representation questionnaires, and 20 learning history interviews (with a total duration of 13 hours).

Section 4.1 reports findings from task performance, Section 4.2 reports findings from differences in interaction strategies, Section 4.3 reports findings from verbal protocols, and Section 4.4 reports findings from the interviews. An alpha value of 0.05 (95% confidence interval) was used for significance testing unless otherwise specified.

4.1 Task performance

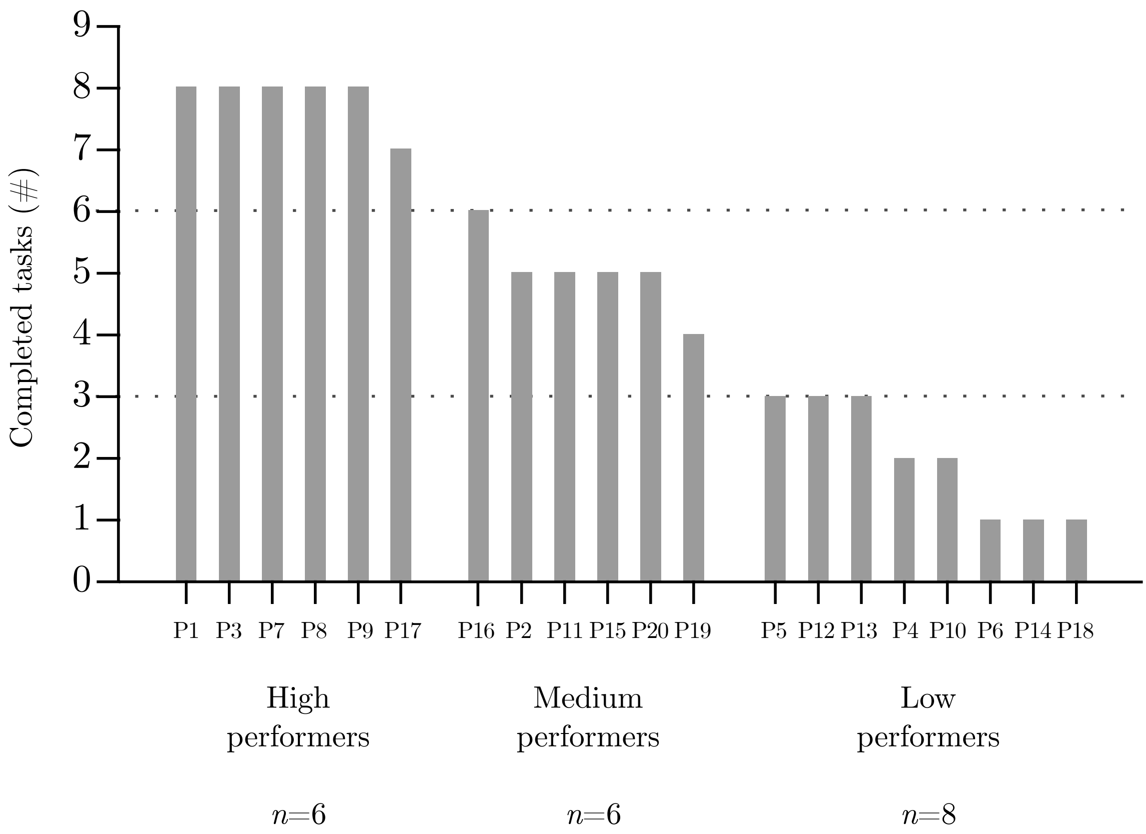
Task performance was analyzed in terms of success in the task (Section 4.1.1), anticipated success rate (Section 4.1.2), completion times (Section 4.1.3), perceived task representativeness (Section 4.1.4), and duration of the subject's professional experience (Section 4.1.5).

4.1.1 Task success

In total, 180 attempts to perform the system administration tasks were observed. Of these attempts, 93 were successful and 87 were unsuccessful. The average success rate was 52%, meaning a subject completed an average of 4.7 ($M = 5$, $SD = 2.6$) out of nine tasks. Although the subjects were experienced professionals with an average of more than ten years of professional system administration experience, the individual differences in task success were significant. The top five subjects completed eight tasks (out of nine), and the bottom five subjects completed two tasks or fewer. The other 10 subjects completed three to seven tasks.

To focus on the differences, the subjects were divided into three *performance groups* based on their task success: *High performers* (who completed 7–9 tasks; $n = 6$), *medium performers* (who completed 4–6 tasks; $n = 6$), and *low performers* (who completed 1–3 tasks; $n = 8$). Subjects' distribution into groups is illustrated in Figure 4.1.

Figure 4.1: Performance groups



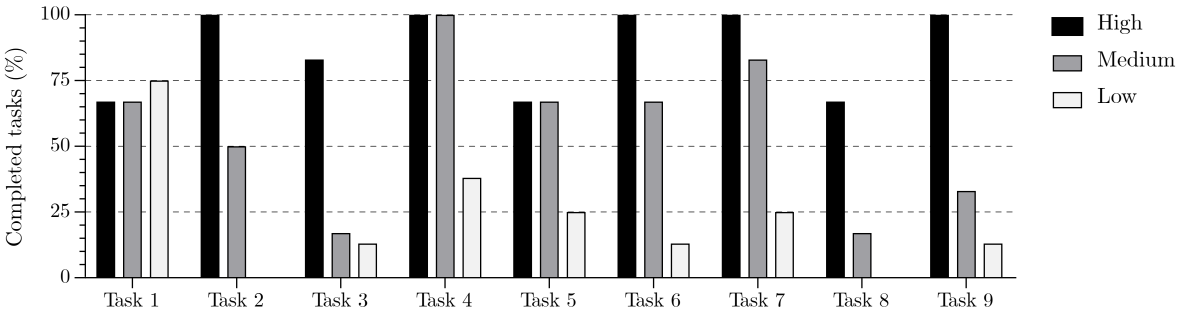
From now on, these performance groups (Figure 4.1) are used to compare the groups' performance, behavior, cognitive outcomes, and learning histories. The group structure and naming conventions follow those used in Programme for International Student Assessment (PISA) studies (OECD, 2018).

High performers successfully completed an average of 7.83 (M = 8, SD = 0.41) tasks, medium performers 5 (M = 5, SD = 0.63) tasks, and low performers 2 (M = 2, SD = 0.93) tasks.

A one-way ANOVA showed significant differences between the groups; $F(2, 17) = 113.3, p < .001$. Post hoc tests with Tukey's honest significant difference (HSD) test showed significant differences between high performers and medium performers (M = 2.833, 95%, CI 1.766–3.901), medium performers and low performers (M = 3, 95%, CI 2.001–3.999), and between high performers and low performers (M = 5.833, 95%, CI 4.835–6.832).

Task-specific differences are shown in Figure 4.2.

Figure 4.2: Task-specific task success percentages per performance group.



The task success data showed that two most manageable tasks were Task 4 (“Write Hello World program and compile it”) with an average completion rate of 75% (15 of 20 participants succeeded) and Task 1 (“Enable firewall and allow SSH access”) with an average completion rate of 70% (14 of 20 participants succeeded).

The two most challenging tasks were Task 8 (“Set up the client to mount home directories from the server”) with an average completion rate of 25% (five of 20 participants succeeded) and Task 3 (“Enable Internet sharing”) with an average completion rate of 30% (seven of 20 participants succeeded). The difference between the performance groups is highlighted in Task 2 (”Configure DHCP server”), where all the high performers (six) completed the task successfully with some time left upon completion (138 seconds on average). In contrast, all the low performers (eight) failed the task.

Another task worth mentioning is Task 3 (“Enable Internet sharing”), which showed a high anticipation rate (0.68) but a low success rate (0.35). The subjects who finished Task 3 did so with plenty of time left upon completion (296 seconds on average).

Task 1 was exceptional; it was the only task in which the low performers performed better than the other groups. Task 1 also had the second-highest success rate of all tasks, and it was the first task in the experiment for all the subjects.

The above results show that the high performers were able to exhibit performance that was reliably superior to that of the other groups.

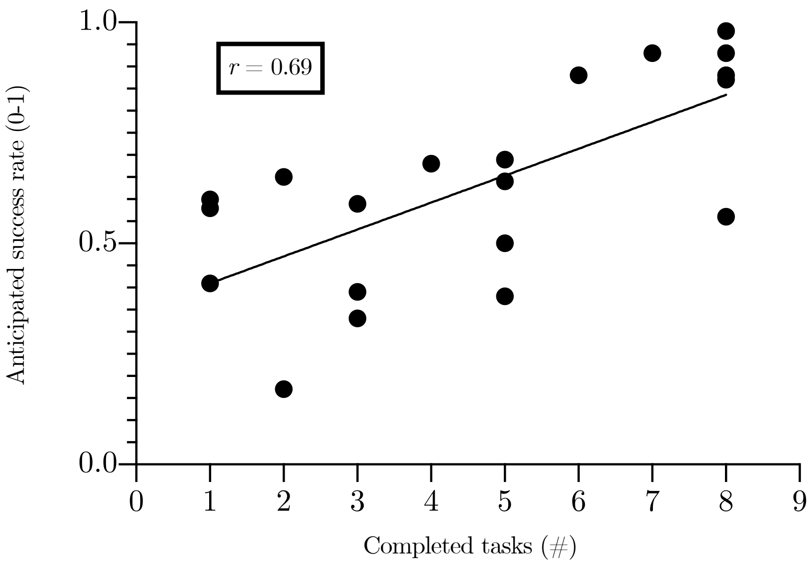
All the task success data is available in Appendix C.1. This shows task success in each task and a given time per subject.

4.1.2 Anticipated success rate

After being made aware of the task assignment, each participant was asked to give their anticipated success rate — their perception of the probability of success in a given time — on a scale of 0 (0%) to 1 (100%) before starting the execution.

The average anticipated success rate was 63%, which was relatively close to the actual success rate (52%). The results of the Pearson correlation indicated that there was a significant and fairly strong positive association between task success and anticipated success rate; $r(18) = .6888, p < .001$. However, there were individuals whose predictions were optimistic and some who were pessimistic. Correlation between completed tasks and anticipated success rate (in scale 0 to 1) is visualized in Figure 4.3.

Figure 4.3: Correlation completed tasks × anticipated success rate.



Note. Correlation coeffiecient ("r") ranges from -1.0 to 1.0. The closer r is to 1 or -1, the more closely the two variables are related.

High performers' average anticipated success rate was 0.86 ($SD = 0.20$), medium performers' average anticipated success rate was 0.63 ($SD = 0.25$), and low performers' average anticipated success rate was 0.47 ($SD = 0.30$). A one-way ANOVA showed significant differences between the groups; $F(2, 17) = 36.34, p < .001$. Post hoc tests with Tukey's HSD showed a significant difference between high performers and medium performers ($M = 0.230$, 95%, CI 0.113–0.346), medium performers and low performers

(M = 0.163, 95%, CI 0.054–0.272), and between high performers and low performers (M = 0.393, 95%, CI 0.284–0.502).

The average success rate for high performers was almost the same (87%) as their anticipated success rate (86%). Medium performers were more optimistic and showed a more considerable difference between actual success rates (56%) and anticipated success rates (63%). Low performers showed an even more substantial difference (22% versus 47%).

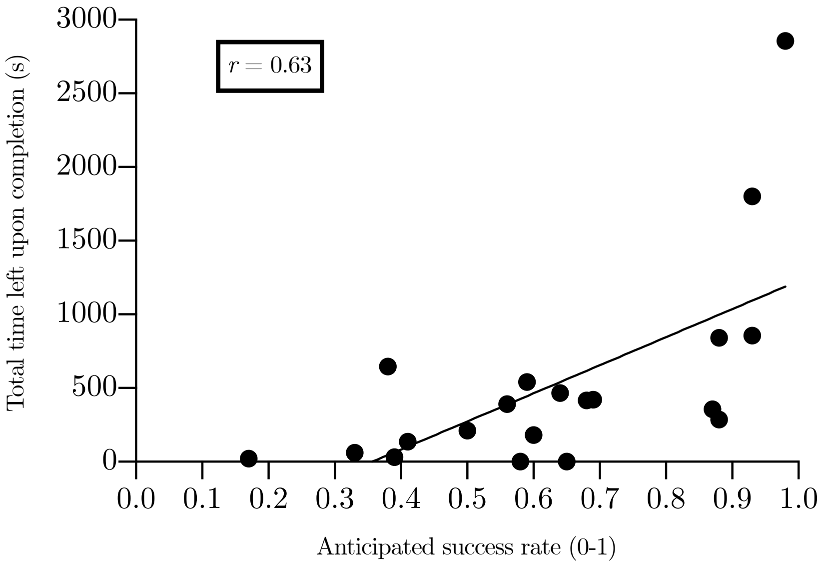
From the above results, we can reliably conclude that high performing subjects were able to predict their success very accurately. The less successful the subjects were, the less accurately and more optimistically they predicted their chances of success.

All collected anticipated success rate data is available in Appendix C.2.

4.1.3 Task completion times

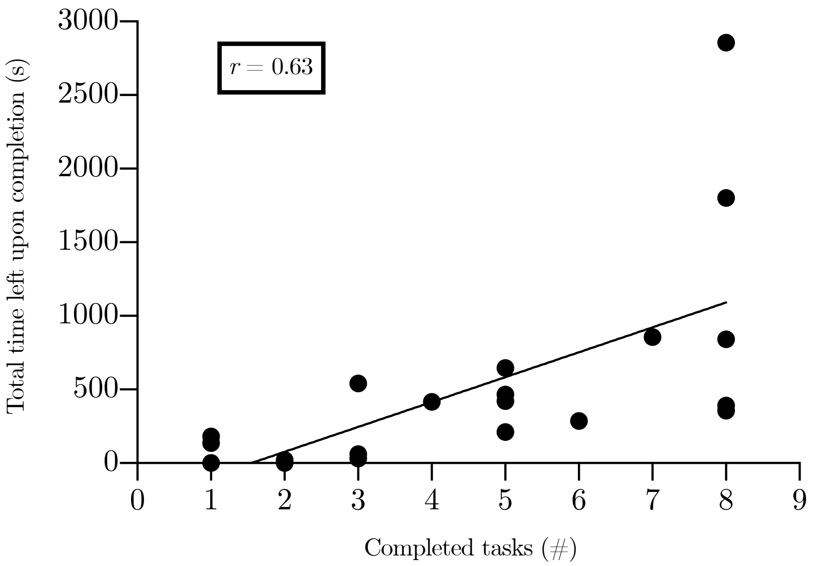
Each task had its unique threshold time of five, 10, or 15 minutes. The total time available for all the tasks was 90 minutes. The average total time left upon completion, which was calculated only for successful trials, was 525 seconds (8 minutes and 45 seconds); an average of 58 seconds per task. Due to the distribution of task success, and the fact that each task had its unique threshold time, the time-left-at-completion metric is most appropriate when totaled over a participant. Results of the Pearson correlation indicated that there was a moderate positive relationship ($r(18) = .6303, p = .003$) between anticipated success and total time left upon completion (visualized in Figure 4.4).

Figure 4.4: Correlation: Anticipated success rate × total time left upon completion.



A reasonably strong correlation ($r(18) = .6323, p = .003$) was also be found between the number of tasks completed and the total time left (visualized in Figure 4.5). As this figure shows, the best performers (measured by the number of tasks completed) were also the fastest The top performer alone had more than twice as much time (47 minutes 35 seconds) remaining after task completion than all the medium performers and low performers combined (17 minutes and 14 seconds).

Figure 4.5: Correlation: Completed tasks × total time left upon completion.



The difference between the high performer group and the other groups was significant. High performers' average total time left at completion was 1183 seconds (19 minutes and 43 seconds), medium performers' average total time left at completion was 83 seconds (1 minute and 23 seconds), and low performers' average total time left at completion was 67 seconds (1 minute and 7 seconds). A one-way ANOVA showed significant differences between the groups; $F(2, 17) = 9.201, p = .002$. Post hoc tests with Tukey's HSD showed significant differences between high performers and medium performers ($M = 1099$, 95%, CI 315–1884) and between high performers and low performers ($M = 1116$, 95%, CI 382–1849). The difference between medium performers and low performers was not significant.

Based on the task completion data, we can reliably conclude that subjects in high performers were not only more successful in completing the tasks, but were also significantly faster than other participants.

All the task completion time data is available in Appendix C.3.

4.1.4 Perceived task representativeness

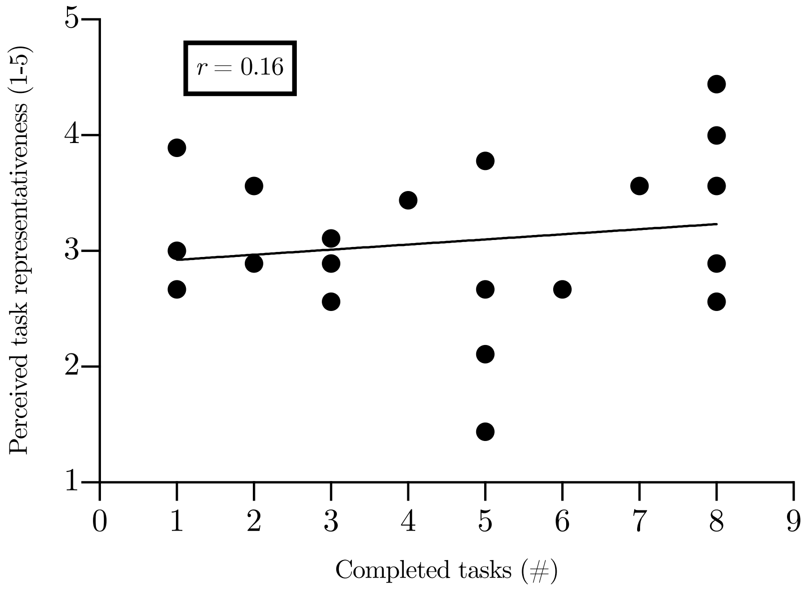
To gain an indicative understanding of the generalizability of results, participants were asked to indicate how well the tasks represented their real-life job duties using a Likert scale of 1 to 5 (1 = strongly disagree, 5 = strongly agree).

The highest grade was given to Task 7 ("Write a shell script to change file access permissions") which received a rating of 4.3 ($M = 5$, $SD = 0.92$) and Task 1 ("Enable firewall and allow SSH access") which received a rating of 4.15 ($M = 4$, $SD = 0.75$).

The lowest grade was given to Task 5 ("Create a software package in deb format") which received a rating of 2.15 ($M = 2$, $SD = 1.27$) and Task 3 ("Enable Internet sharing") which received a rating of 2.20 ($M = 2$, $SD = 1.11$).

Non-existing correlation ($r(18) = .1614, p = .497$) between the number of completed tasks and perceived task representativeness (on a scale of 1 to 5) is visualized in Figure 4.6.

Figure 4.6: Correlation: Completed tasks × perceived task representativeness.



Note. No relationship means, that as one value increases, there is no tendency for the other value to change in a specific direction.

When comparing performance groups, the high performers' average perceived task representativeness (on a scale of 1 to 5) was 3.5 (SD = 1.36), the medium performers' was 2.69 (SD = 1.36), and the low performers' was 3.07 (SD = 1.38). A Kruskal-Wallis H test showed that there was a statistically significant difference between the groups; $\chi^2 = 9.428, p = .009$. The Kruskal-Wallis test is a version of the independent measures ANOVA that can be performed on ordinal data such as Likert scale answers. The results of the Bonferroni post hoc test showed a significant difference between the high performers and the medium performers.

We can conclude that there was no significant correlation between task performance and perceived task representativeness. Success in the tasks was not determined by how well the assignments matched the job duties. However, when comparing the groups, the tasks represented the actual daily routines of the high performers significantly better than they represented those of the medium performers.

All the collected task representativeness data is available in Appendix C.4

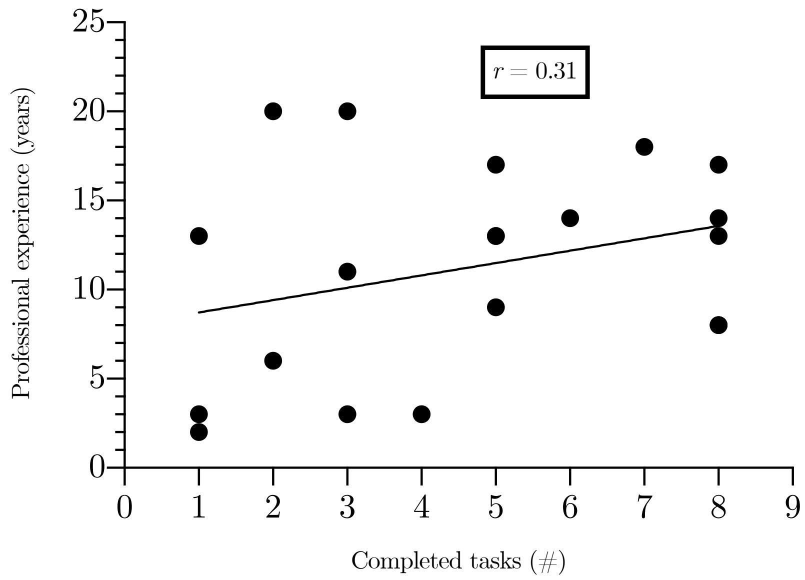
4.1.5 Duration of experience

Finally, the relationship between the duration of professional system administration experience and task performance was examined.

On average, participants had 12 years (M = 13, SD = 5.8) years of professional experience as system administrators.

Weak correlation ($r(18) = .3094, p = .184$) between the number of completed tasks and years of professional system administration experience is visualized in Figure 4.7.

Figure 4.7: Correlation: Completed tasks × years of professional experience.



When comparing performance groups, the high performers had an average of 13 (M = 13.5, SD = 4.29) years of experience, the medium performers 11.5 (M = 13, SD = 4.89) an average of years of experience, and the low performers an average of 9.75 (M = 8.5, SD = 7.44) years of experience. A one-way ANOVA showed that there was no significant difference between the groups; $F(2, 17) = 0.5215, p = .603$.

In conclusion, the results show weak relationship between length of experience and task success. However, with five or less years of system administration experience, the best score was four completed tasks, and even 20 years of experience did not guarantee success.

4.2 Task solution strategies

The actions taken by the subjects during task performance were analyzed from the video and divided into the categories described in Table 3.10. The proportions of categorized actions are presented in Table 4.1.

	Performance group					
Action	**High**	**Medium**	**Low**	$\boldsymbol{F}$	$\boldsymbol{\eta^2}$	$\boldsymbol{p}$
Mean reached sub-goals	26.83	15.33	7.25	404.1	0.9794	<.001
Directly useful action %	49	29	15			
Mean	31	17.7	10	272.9	0.9698	<.001
Useless action %	8	17	27			
Mean	5.3	10	18.3	21.69	0.7185	<.001
Harmful action %	1	3	6			
Mean	0.7	2	3.9	7.683	0.4747	.004
Information search: system %	11	21	14			
Mean	6.7	12.5	9.8	5.630	0.3984	.013
Information search: Internet %	16	17	27			
Mean	10	10.2	18.8	36.35	0.8105	<.001
Testing %	15	13	11			
Mean	9.3	7.8	7.6	0.5153	0.05716	.606
Total %	*100*	*100*	*100*			

Note. Significant at the $p < 0.05$ level.
$F = F$ ratio, $\eta^2 =$ effect size, $p =$ significance.

Table 4.1: Actions taken during task performance in percentages and as comparisons of means.

The high performers took an average of 63 ($M = 61$, $SD = 4.7$) actions, the medium performers took an average of 60.2 ($M = 60.5$, $SD = 4$) actions, and the low performers took an average of 68.3 ($M = 69.5$, $SD = 8$) actions. As expected, the high performers reached significantly more sub-goals than other groups, and the medium performers reached the sub-goals at more than double the rate of the low performers.

About half (49%) of all actions taken by the high performers were useful for the completion of the task and led toward a successful solution. In addition to directly useful actions, the average task performance for high-performing subjects consisted of accurate Internet searches (11% of all actions) and testing a workable solution (15% of all actions).

A third (33%) of all actions taken by the low performers were either of no use or were harmful to the success of the task. In addition to non-beneficial actions, a large proportion of the actions of lower-performing subjects consisted of searching for information (41%). The video analysis revealed that high-performing subjects spent less time on searching and their search keywords were more specific. They used syntactic details such as “deb packages file structure”, whereas lower-performing subjects searched for higher-level concepts like “shell scripting if-else” and for fully functional example solutions.

In conclusion, a significantly smaller proportion of the high performers’ actions were information searches (27% versus 38% versus 41%), and a significantly higher proportion of actions were useful (49% versus 29% versus 15%). Low performers took many actions without compensation.

4.3 Verbal protocols

This section examines the verbal protocols of subjects during both the anticipation of tasks (Section 4.3.1) and the performance of tasks (Section 4.3.2).

Subjects’ verbal protocols were analyzed from the video recordings.

4.3.1 Depth of anticipation

Pre-task anticipations were analyzed in terms of duration, correct steps mentioned, and defined cognitive actions. The average duration, number of correct steps mentioned, anticipated success rate, and the think-aloud data during anticipation is presented in Table 4.2.

Variable	Performance group High	Medium	Low	F	η^2	p
Mean average duration (s)	75	69	66	2.229	0.2077	.138
Mean correct steps mentioned	16.8	10.2	3.3	53.57	0.8631	<.001
Mean anticipated success rate %	86	63	47	36.34	0.2911	<.001
Confidence %	61	44	34			
Mean	16.8	11.3	6.9	34.39	0.8018	<.001
Unconfidence %	11	35	51			
Mean	3	9	10.4	24.31	0.7410	<.001
Planning %	28	21	15			
Mean	7.8	5.5	3.1	14.39	0.6287	<.001
Total %	*100*	*100*	*100*			

Note. Significant at the $p < 0.05$ level.

F = F ratio, η^2 = effect size, p = significance.

Table 4.2: Analysis of pre-task anticipations in percentages and as comparisons of means.

The results show that the difference between the groups in the duration of the anticipation was not significant. However, high performers mentioned significantly more correct steps ($\bar{x} = 16.8$) than middle performers ($\bar{x} = 10.2$) and low performers ($\bar{x} = 3.3$).

Video analysis revealed that, on many occasions, high performers were able to announce the exact commands they were going to execute to complete the task. In contrast, low performers frequently stated that they did not yet have a plan for how to solve the task, but thought they would find instructions on Google and proceed accordingly.

As expected, the high performers were more confident during task anticipation. As stated earlier, high performers predicted their success very accurately. Medium performers and low performers were more optimistic (and even unnecessarily confident) in relation to their actual performance. In summary, the better a participant performed at the tasks, the more confident they were likely to be in the anticipation phase.

4.3.2 Cognitive outcomes

Think-aloud data during task performance is presented in Table 4.3.

	Performance group					
Category	**High**	**Medium**	**Low**	F	η^2	p
Confidence %	59	40	21			
Mean	16	8.8	4.9	50.63	0.8562	<.001
Unconfidence %	12	31	64			
Mean	3.2	6.8	15	26.38	0.7563	<.001
Planning %	29	29	15			
Mean	8	6.5	3.6	20.20	0.7039	<.001
Total %	*100*	*100*	*100*			

Note. Significant at the $p < 0.05$ level.

$F = F$ ratio, η^2 = effect size, p = significance.

Table 4.3: Analysis of think-aloud protocols in percentages and as comparisons of means.

Considering the significant differences in task performance, it was expected that high performing subjects would express more confidence during the performance of the tasks than the lower-performing subjects.

Low performers spent more time gaining a comprehensive understanding of the current state of the system and the difference between this state and the intended goal. Their solution paths included lots of loops, including information retrieval, and trial and error experiments. This approach produced many statements that were interpreted as unconfidence (e.g., “I have no idea why this doesn’t work”).

High performers were more often on the right track from the beginning; their focus was on details (e.g., parameters of specific command), and therefore their cognitive outcomes were more often interpreted as confidence (e.g., “Yes, now this works, now we need only...”). In conclusion, the better the subjects performed, the more positive their cognitive outcomes.

4.4 Learning histories

Subjects’ learning histories are described in Sections 4.4.1, 4.4.2, and 4.4.3.

4.4.1 Skill acquisition

Computer-related activities before professional career

Subjects were asked to list computer-related activities that they were involved before their professional career in the field, which they continued to do for an hour or more per week or which they engaged in for more than a total of 50 hours. Subjects listed weekly hours, years, duration in years, utility, and effort for each given computer-related activity.

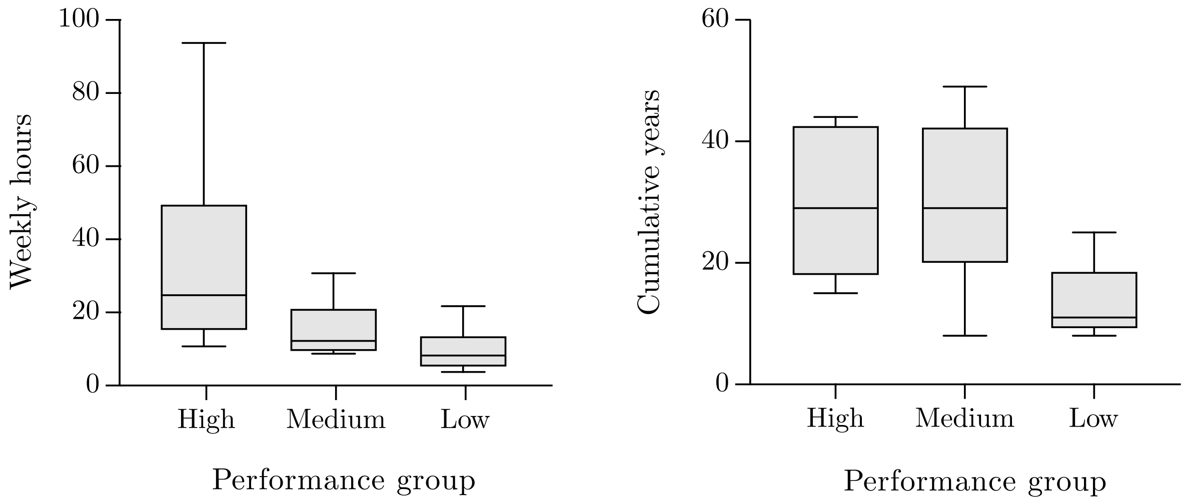
Subjects reported spending an average of 18 hours ($SD = 19.8$) per week on computer-related activities before their professional career. The most weekly hours were reported as being spent on gaming (5.15 hours per week). Measured in years, the most popular activity was also gaming (with an average time of 6.8 years).

High performers reported spending an average of 33.50 hours ($SD = 30.68$) per week, and engaging in computer-related activities during an average of 29.67 years ($SD = 12.77$). Medium performers reported spending an average of 14.50 hours ($SD = 8.22$) per week, during an average of 29.83 years ($SD = 14.25$). Low performers reported spending an average of 9 hours ($SD = 5.90$) per week, during an average of 13.75 years ($SD = 5.99$).

In weekly hours there were no statistically significant differences between group means as determined by one-way ANOVA; $F(2, 17) = 3.476, p = .054$. However, post hoc tests with Tukey's HSD showed a significant difference between high performers and low performers ($M = 24.50$, 95%, CI 0.06268-48.94). In accumulated years there was a statistically significant difference between groups as determined by one-way ANOVA; $F(2, 17) = 5.018, p = .019$. Post hoc tests with Tukey's HSD showed a significant difference between high performers and low performers ($M = 15.92$, 95% CI 0.5856-31.25), and between medium performers and low performers ($M = 16.08$, 95% CI 0.7523-31.41).

The differences between the performance groups in weekly hours and cumulative years are shown in Figure 4.8.

Figure 4.8: Box plot illustrating the time spent on computer-related activities before a professional career. Vertical bars denote 95% confidence intervals (CIs).

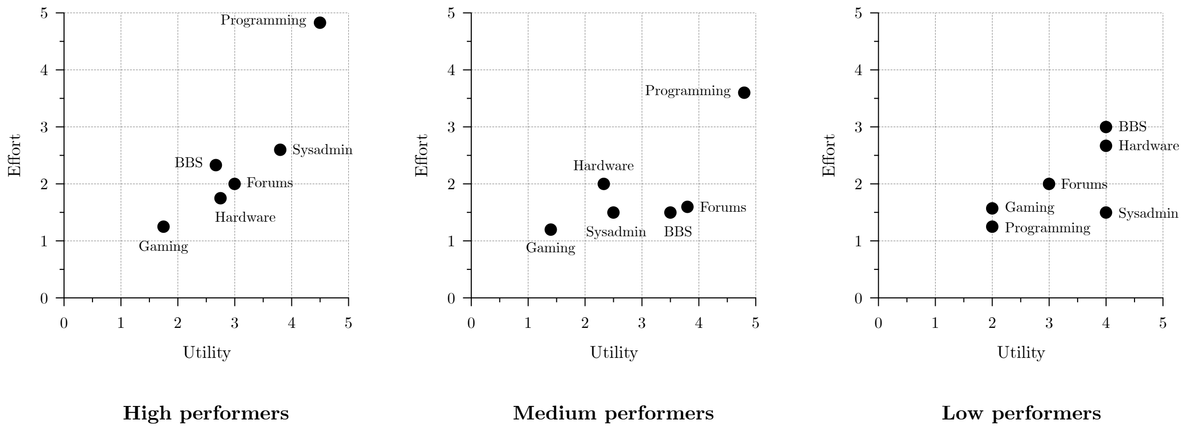


There were significant differences between the activities of the performance groups. Among the high performers, the most popular activity before a professional career was programming, with 9.50 (SD = 5.6) weekly hours. Among the low performers, the top activity was gaming, with 2.9 (SD = 2.2) weekly hours. High performers found programming to be by far the most useful activity in terms of current skills (utility; $\bar{x}$ = 4.5) and the most demanding activity (effort; $\bar{x}$ = 4.83).

Medium performers found the programming activities even more useful (utility; $\bar{x}$ = 4.8) and demanding (effort; $\bar{x}$ = 3.6). Surprisingly, the low performers considered programming to be the least valuable (utility; $\bar{x}$ = 2.0) and least demanding (effort; $\bar{x}$ = 1.25) of their activities.

The differences between the performance groups in relation to the perceived effort and utility of different computer-related activities is shown in Figure 4.9.

Figure 4.9: XY graph of perceived utility and effort of computer-related activities before professional career per performance group.



The time spent, utility, and effort perceived by the subjects in relation to various computer-related activities before their professional career, and the differences between the performance groups, are shown in Table 4.4. A Kruskal-Wallis H test was performed to explore the differences between the groups (alpha value 0.05).

Activity	High $\bar{x}$	High SD	Medium $\bar{x}$	Medium SD	Low $\bar{x}$	Low SD	df	χ^2	p
Gaming									
Hours/week	8.17	9.39	5.17	7.44	2.88	2.23	2	0.4559	.811
Years	6.33	4.93	8.50	6.41	5.88	4.05	2	1.012	.617
Utility (1-5)	1.75	0.96	1.40	0.55	2.00	1.26	2	0.3247	.872
Effort (1-5)	1.25	0.50	1.20	0.45	1.57	0.79	2	0.9732	.657
Programming									
Hours per week	9.50	5.61	3.50	3.45	1.00	1.69	2	11.70	$< .001$
Years	7.67	1.75	6.17	4.12	2.38	3.34	2	6.657	.029
Utility (1-5)	4.50	1.22	4.80	0.45	2.00	0.00	2	9.111	.008
Effort (1-5)	4.83	0.41	3.60	1.34	1.25	0.50	2	10.27	.001
BBS									
Hours per week	6.86	16.03	2.17	2.04	0.75	1.16	2	1.819	.420
Years	1.00	1.26	2.00	1.90	0.63	1.06	2	2.285	.336
Utility (1-5)	2.67	0.58	3.50	0.58	4.00	1.41	2	2.814	.374
Effort (1-5)	2.33	1.15	1.50	0.58	3.00	1.73	2	2.530	.377
Tech forums									
Hours per week	3.33	3.78	1.50	1.22	0.63	0.92	2	3.652	.159
Years	4.17	3.66	5.50	3.33	1.00	2.14	2	5.974	.044
Utility (1-5)	3.00	0.00	3.80	0.84	3.00	0.00	2	3.108	.232
Effort (1-5)	2.00	0.82	1.60	0.55	2.00	0.00	2	1.156	.576
Hardware tweaking									
Hours per week	1.50	1.22	0.67	0.82	0.67	1.16	2	1.907	.403
Years	3.83	3.54	3.33	4.80	1.00	1.51	2	2.153	.353
Utility (1-5)	2.75	0.50	2.33	0.58	4.00	1.00	2	5.000	.069
Effort (1-5)	1.75	0.50	2.00	0.00	2.67	2.08	2	0.4410	$> .999$
Personal Sysadmin									
Hours per week	3.17	3.31	1.17	1.60	2.75	6.98	2	3.349	.189
Years	5.00	3.58	2.50	2.95	2.38	3.85	2	2.320	.328
Utility (1-5)	3.80	1.10	2.50	0.71	4.00	1.73	2	1.954	.505
Effort (1-5)	2.60	1.82	1.50	0.71	1.50	1.29	2	0.8920	.699

Note. Significant at the $p < 0.05$ level.
$\bar{x}$ = Mean, SD = standard deviation, df = degrees of freedom,
χ^2 = Chi-square value, p = significance.

Table 4.4: Reported computer-related activities before a professional career.

In summary, high performers had previously focused on demanding and constructive activities such as programming and hosting dial-up bulletin board systems (BBS). They also felt that these activities were beneficial for their current skills. Low performers, on the other hand, had spent the most time on consuming activities (i.e., computer games), which they did not find to be demanding or beneficial for their current skills.

Computer-related activities during professional career

Subjects were asked to list computer-related activities that they had been involved during their professional career, which they been doing for an hour or more per week or which they engaged in for more than a total of 50 hours. Subjects listed weekly hours, years, duration in years, utility, and effort for each given computer-related activity.

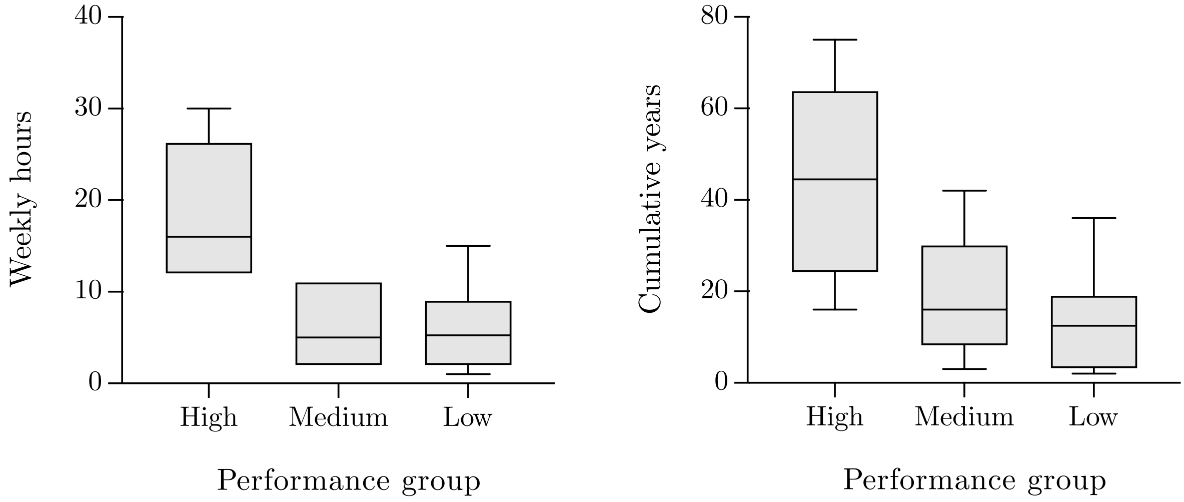
Subjects reported spending an average of 9.7 (SD = 7.9) hours per week in their free time on computer-related activities. Cumulative years of experience accumulated during their career, across various activities, were reported to be 24.3 (SD = 20.3) years. By far the most popular activity was technology forums, with 4.6 (SD = 5.37) weekly hours.

High performers reported spending an average of 18.50 hours (SD = 7.37) per week, and engaging in computer-related activities during an average of 44.50 accumulated years (SD = 21.79), during their professional career. Medium performers reported spending an average of 6 hours (SD = 4.52) per week, during an average of 18.83 years (SD = 14.18). Low performers reported spending an average of 5.81 hours (SD = 4.71) per week, during an average of 13.25 years (SD = 11.35).

In weekly hours there was statistically significant differences between group means as determined by one-way ANOVA; $F(2, 17) = 10.74, p < .001$. Post hoc tests with Tukey's HSD showed a significant difference between high performers and medium performers (M = 12.50, 95%, CI 4.242-20.76), and between high performers low performers (M = 12.69, 95%, CI 4.963-20.41). Also, in accumulated years there was a statistically significant difference between groups as determined by one-way ANOVA; $F(2, 17) = 7.158, p = .006$. Again, post hoc tests with Tukey's HSD showed a significant difference between high performers and medium performers (M = 25.67, 95% CI 2.166-49.17), and between high performers and low performers (M = 31.25, 95% CI 9.267-53.23).

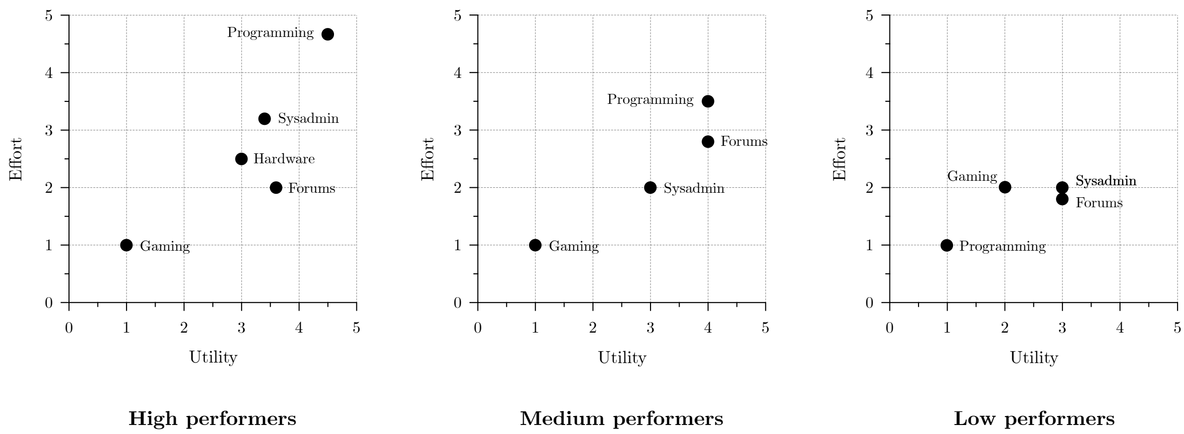
The differences between the performance groups in weekly hours and cumulative years are shown in Figure 4.10.

Figure 4.10: Time spent on computer-related activities during professional career box plot. Vertical bars denote 95% confidence intervals (CIs).



The differences between the performance groups in relation to the perceived effort and utility of different computer-related activities is shown in Figure 4.11. Again, the same pattern in relation to perceived utility and effort associated programming can be observed. High performers and middle performers found programming to be a useful and demanding activity, but low performers did not.

Figure 4.11: XY graph of perceived utility and effort of computer-related activities during professional career per performance group.



The time spent, utility, and effort perceived by the subjects on various computer-related activities during their professional career, and the differences between the performance groups, are shown in Table 4.5. A Kruskal-Wallis H test was performed to explore the differences between the groups (alpha value 0.05).

	High		Medium		Low				
Activity	$\bar{x}$	*SD*	$\bar{x}$	*SD*	$\bar{x}$	*SD*	*df*	χ^2	*p*
Gaming									
Hours/week	2.50	4.18	0.67	1.63	0.25	0.71	2	1.446	.520
Years	5.00	7.75	2.33	5.72	0.63	1.77	2	1.449	.520
Utility (1-5)	1.00	0.00	1.00	0.00	2.00	0.00	2	3.000	.500
Effort (1-5)	1.00	0.00	1.00	0.00	2.00	0.00	2	3.000	.500
Programming									
Hours per week	4.83	4.02	2.00	2.25	0.19	0.53	2	10.91	.002
Years	12.00	3.35	6.33	7.00	0.50	1.41	2	10.82	.002
Utility (1-5)	4.50	0.84	4.00	0.00	1.00	0.00	2	3.623	.135
Effort (1-5)	4.67	0.52	3.50	0.71	1.00	0.00	2	5.233	.024
Tech forums									
Hours per week	8.00	6.57	3.33	3.50	3.00	4.96	2	3.610	.167
Years	11.67	6.59	8.50	5.68	8.25	7.80	2	1.173	.575
Utility (1-5)	3.60	0.89	4.00	1.00	3.00	0.00	2	1.768	.454
Effort (1-5)	2.00	0.71	2.80	1.48	1.80	0.84	2	1.862	.438
Hardware tweaking									
Hours per week	0.50	0.84	0.00	0.00	0.00	0.00	2	4.912	.158
Years	4.00	6.48	0.00	0.00	0.00	0.00	2	4.912	.158
Utility (1-5)	3.00	0.00	–	–	–	–			
Effort (1-5)	2.50	0.71	–	–	–	–			
Personal sys. admin.									
Hours per week	2.67	3.67	0.33	0.82	2.38	3.58	2	3.788	.157
Years	11.83	6.49	1.67	4.08	3.88	5.96	2	6.349	.037
Utility (1-5)	3.40	1.67	3.00	0.00	3.00	0.00	2	0.4167	.833
Effort (1-5)	3.20	1.48	2.00	0.00	2.00	0.82	2	2.263	.390

Note. Significant at the $p < 0.05$ level.
$\bar{x}$ = Mean, SD = standard deviation, df = degrees of freedom,
χ^2 = Chi-square value, p = significance.

Table 4.5: Computer-related activities during professional career.

In conclusion, the results are aligned with the previous section, but the differences between groups are even more considerable. The better the group performed the tasks, the more likely they were to spend their free time on computer-related activities. Once again, programming stands out as an exceptional activity that sets excellent performers apart.

Origin of knowledge

Subjects were asked how much of their knowledge about the system they currently supervised was due to hands-on experience (with the current system and other systems), formal training programs (e.g., Microsoft, Sun, Novell), reading or research on their own (e.g., books and the Internet), relevant formal education (e.g., university studies), and working with and learning from others.

The answers per performance group are shown in Table 4.6.

	High		**Medium**		**Low**	
Origin of knowledge	$\bar{x}$ (%)	***SD***	$\bar{x}$ (%)	***SD***	$\bar{x}$ (%)	***SD***
Hands-on experience	51	15	58	22	49	26
Formal training programs	–	–	5	5	–	–
Reading and research on your own	27	13	21	18	27	28
Relevant formal education	5	8	7	11	5	8
Working with and learning from others	14	10	11	12	19	15

Table 4.6: Origin of the knowledge.

There were no significant differences between performance groups. All the performance groups agreed that the most critical source of their knowledge was hands-on experience, the second most important was self-study, and the third most important was working with and learning from others. Formal education or training programs were not considered to be a relevant source of current knowledge.

Accumulated duration of practice

Accumulated duration of practice is a sum of any type of computer-related activity reported by a subject, including ICT professional experience over the years, and all the computer-related activities both before and during the professional career. However, format education is not included since the data on duration of studies were not collected.

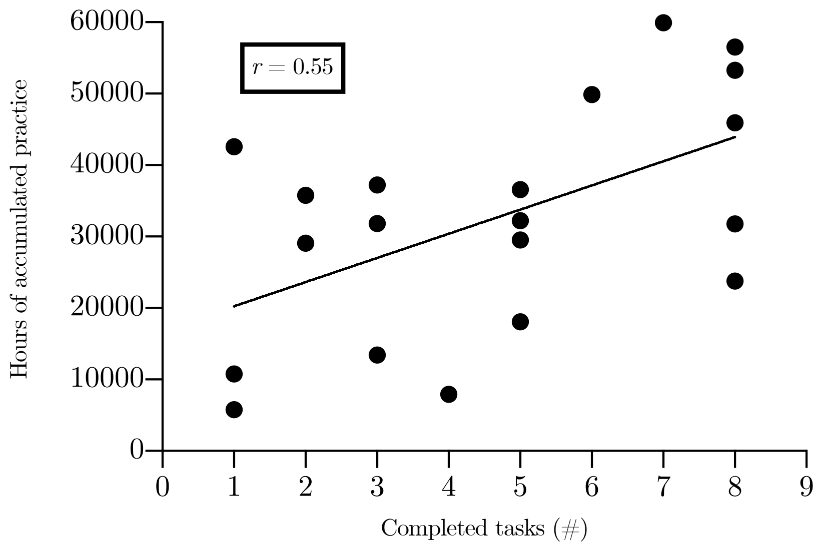
Professional experience has been converted into hours of practice by multiplying the years of experience with an average *annual hours actually worked per employee* in Finland in 2011, which was 1584 hours (Official Statistics of Finland, 2020).

An average accumulated duration of practice over the years was 33395 hours ($M = 32028, SD = 17737$). High performers' average accumulated duration was 47877 hours ($M = 49603, SD = 18632$), medium performers' 29036 hours ($M = 30885, SD = 14611$), and low performers' 25802 hours ($M = 30444, SD = 13840$).

In accumulated duration of practice there was statistically significant differences between group means as determined by one-way ANOVA; $F(2, 17) = 3.761, p = .044$. Post hoc tests with Tukey's HSD showed a significant difference between high performers and low performers ($M = 22075$, 95% CI 444-43706). The difference between high performers and medium performers and the difference between medium performers and low performers were not significant.

The results of the Pearson correlation indicated that there was a significant and moderate positive association between the number of completed tasks and accumulated duration of practice; $r(18) = .5458, p = .013$. The correlation is visualized in Figure 4.12.

Figure 4.12: Correlation: Completed tasks × Hours of accumulated practice.



4.4.2 Current system environment

Origin of the current system

The subjects were asked whether the system they currently supervised was a) mostly self-built, b) partly self-built and partly inherited, or c) mostly inherited. The distribution of subjects' responses is shown in Table 4.7.

Origin of current system	**High**	**Medium**	**Low**
Mostly self-built	66.6% (4 of 6)	50% (3 of 6)	0% (0 of 8)
Partly self-built and partly inherited	33.3% (2 of 6)	33.3% (2 of 6)	37.5% (3 of 8)
Mostly inherited	0% (0 of 6)	16.7% (1 of 6)	62.5% (5 of 8)
Total	*100% (6 of 6)*	*100% (6 of 6)*	*100% (8 of 8)*

Table 4.7: Origin of the current system.

High performers and medium performers were more likely to have built the system that they administrated, whereas the low-performing subjects were mostly administrating a system they had inherited.

Uniqueness of current system

To gain understanding about the generalizability of the findings, the subjects were asked whether the system they currently supervised was either a) unlike any other system in the world, b) unusual for the most part, c) equal parts unusual and generic, d) generic for the most part, or e) completely standard.

The distribution of subjects' responses is shown in Table 4.8.

Uniqueness of current system	High	Medium	Low
Unlike any other system	0% (0 of 6)	0% (0 of 6)	0% (0 of 8)
Mostly unusual	0% (0 of 6)	0% (0 of 6)	0% (0 of 8)
Equal parts unusual and generic	50% (3 of 6)	16.7% (1 of 6)	37.5% (3 of 8)
Mostly generic	50% (3 of 6)	66.6% (4 of 6)	37.5% (3 of 8)
Completely standard	0% (0 of 6)	16.7% (1 of 6)	25.0% (2 of 8)
Total	*100% (6 of 6)*	*100% (6 of 6)*	*100% (8 of 8)*

Table 4.8: Uniqueness of the current system.

There was no noticeable difference between the performance groups regarding the uniqueness of the system. All subjects felt that the system they maintained was somewhat generic and built from standard open-source components. Genericity was also mentioned as an objective because it makes the system easier to understand and maintain.

Working environment: Number of co-workers

Subjects were asked how many co-workers they had working on the same tasks as them. An average of 2.5 people ($M = 2$, $SD = 1.6$) worked on the same tasks as the subjects.

There were no significant differences in the number of co-workers between the performance groups. High performers reported having an average of 2.5 co-workers ($M = 2$, $SD = 2$), medium performers and average of 2.7 co-workers ($M = 2$, $SD = 1.2$), and low performers and average of two co-workers ($M = 2.3$, $SD = 1.7$).

Currently supervised system parts, products, and sub-systems

Participants were asked to list the system parts, products, and sub-systems they currently supervised. The subjects listed their various areas of responsibility. The accuracy of the answers varied from the broad ("servers in general") to the very detailed ("the IPv6 tunnel broker system"). The system parts or roles that received at least three mentions, and the group-specific percentage of mentions, are presented in Table 4.9.

System part / role	High	Medium	Low
Backups & data recovery (%)	17	33	0
Customer support, helpdesk (%)	0	17	63
Cybersecurity (%)	50	0	0
Databases (%)	17	33	0
Linux servers (%)	83	50	25
Networks (%)	33	33	13
Management (%)	50	17	0
Operating systems (%)	33	33	0
System Architecture (%)	33	0	0
Workstations (%)	0	17	75

Table 4.9: Currently supervised system parts. Percentages of mentions per group.

Various system parts and roles were mentioned. Since these systems are not comparable, no significant conclusions can be drawn from the answers. However, some observations can be made from the data. High performers had broad responsibilities, such as system architecture, high-performance cluster development, and development of data center operations. In contrast, low performers seemed to focus on more practical tasks, such as customer support and workstation administration.

4.4.3 Deliberate practice

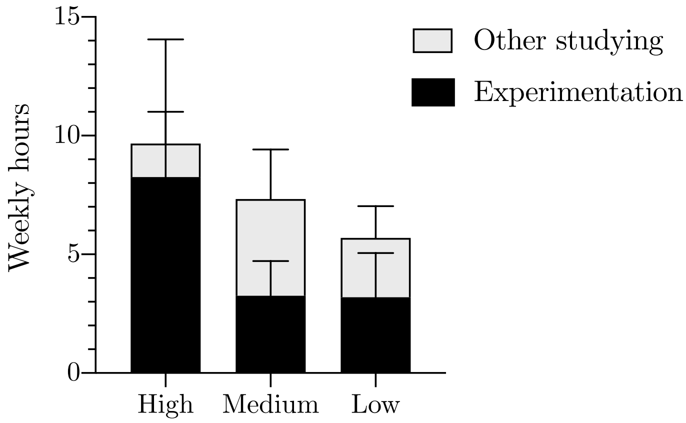
Solitary time studying

Subjects were asked how many hours per week they used for studying when they were adopting a new sub-system, how much time they used for experimenting with how it worked, and whether they had a specific test environment for such experimentation.

Subjects reported spending an average of 7.38 hours ($M = 7$, $SD = 3.26$) per week to adopt a new sub-system that didn't require their full attention, such as a new email server. Of that time, they reported spending an average of 4.73 hours ($M = 3$, $SD = 4$), experimenting with how it worked.

The weekly study and experimentation hours reported by the performance groups are shown in Figure 4.13. A one-way ANOVA shows that there is no statistically significant difference between the groups when looking at the weekly hours spent studying; $F(2,17) = 3.126, p = .070$. However, when looking at the hours spent on experimentation, there is a statistically significant difference between the groups; $F(2,17) = 4.579, p = .026$. Post hoc tests with Tukey's HSD show a significant difference between high performers and low performers (M = 5.063, 95%, CI 0.3376-9.787).

Figure 4.13: Time used for studying per performance group. Vertical bars denote 95% confidence intervals.



All subjects reported using a testing environment to conduct experiments. They reported using both virtual environments and decommissioned production servers for experimentation.

There is no significant difference in the use of time between the performance groups. However, high performers spent a more substantial portion of their time doing experiments.

Problem-solving

Subjects were asked about the actions they took when they had to fix something on the system that they did not already know how to fix.

The results show that the most popular action was searching for information from the Internet ("Do research via the web newsgroups") with an average of 48%. The second most popular action varied by performance group. High performers and middle performers reported that they experimented to see what worked, while low performers reported that they consulted other people. The data is shown in Table 4.10.

Action	High		Medium		Low	
	$\bar{x}$	*SD*	$\bar{x}$	*SD*	$\bar{x}$	*SD*
Consult with people you know who have experience (%)	7	8	11	13	26	19
Contact manufacturer support (%)	5	8	3	6	4	7
Contact third party support (%)	–	–	–	–	–	–
Do research via books or technical literature (%)	–	–	–	–	–	–
Do research via the web newsgroups (%)	45	5	42	26	54	18
Experiment to try and see what works (%)	40	13	18	16	12	15
Use diagnostic tools (%)	3	8	10	8	4	8
Take an action that is not listed above (%)	–	–	–	–	–	–

Note. $\bar{x}$ = mean, SD = standard deviation.

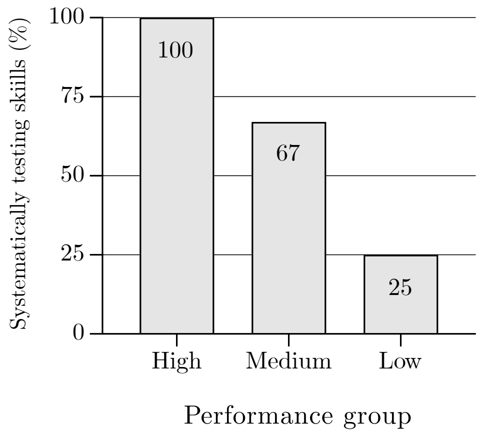
Table 4.10: Actions to be taken when the solution is not known in advance.

In conclusion, the better the subject performed in the tasks, the more likely they were to utilize experimentation when solving problems.

Self-challenge

Finally, subjects were asked whether they felt that they were systematically testing their skills, and to elaborate on what they mean by their answer. The question was binary (yes/no); in reality, the answer may not be so straightforward. Responses such as “rarely, maybe sometimes, if there is time” and “not interested, there are ready-made solutions on the Internet” were interpreted as negative responses. Responses such as “due to my curious nature, I must try” were interpreted as positive responses. The distribution of subjects’ responses is shown in Figure 4.14.

Figure 4.14: Proportion of subjects who reported challenging their own skills.



A total of 60% of the subjects reported that they systematically tested their system administration skills in their daily routine. As expected, 100% of high performer subjects responded that they systematically tested their system administrator skills as part of their daily routines. The corresponding figures were 67% and 25% for medium and low performers, respectively.

The results suggest that the more actively a participant challenged themselves, the better their performance was likely to be.

5 Discussion

This Master's thesis focuses on studying differences in the skill levels of professional system administrators and the factors influencing them. Twenty experienced system administrators participated in the study and were divided into three groups based on their measured performance. This section presents the findings, their implications (Section 5.1), limitations, and ideas for future development (Section 5.2).

5.1 Findings and implications

The findings of the study can be divided into four main themes:

1. *Task performance:* high performers were significantly faster and more successful in completing the tasks. The difference between high performers and medium performers, and between high performers and low performers, was significant in both in task success and in task completion time. No significant differences were observed between medium and low performers in either task success or completion time.

2. *Representation of the problem:* high performers were able to predict their success very accurately. They described the required actions and the details that they needed to resolve the problem notably more accurately and verbalized more thoughts reflecting confidence and effective planning during both task anticipation and task performance.

3. *Practice and learning:* according to the interviews, the high performer group spent considerably more time and effort on computer-related activities (both before and during their professional career) compared to other groups. The most sharply differentiating factors between expert and less-accomplished performers were the quality and quantity of high-effort problem-solving activities, such as programming. Furthermore, when learning and adopting a new sub-system, subjects in the high performer group reported spending significantly more time on experimentation than other groups.

4. *Impact of mere experience:* there was no relationship between the duration of participants' professional experience and task success. However, all the superior performers had at least 10 years of professional experience and intense involvement in constructive computer-related activities prior to their professional careers.

With the results obtained, it is possible to evaluate the research questions that guided the experiment. We wanted to know if professional system administrators were able to exhibit performance reliably superior to others ($RQ1$); if so, what the individual differences were ($RQ2$); if there was a relationship between the duration of professional system administration experience and measured performance ($RQ3$); if there was a relationship between the accumulated duration of practice and measured performance ($RQ4$); if the types of practice professional system administrators had engaged in differed as a function of performance ($RQ5$); and the amount and type of training required by someone hoping to become a high-performing system administrator ($RQ6$).

Regarding $RQ1$, a group of five subjects exhibited performance consistently superior to the others. This differentiation was expected based on the previous research (Ericsson, 2008), as similar differences in performance have been found across a wide variety of domains, including medicine (Ericsson, 2004), sports (Côté et al., 2005; Ericsson, 1996), and games (Ericsson, 1996). Although all subjects had professional experience as system administrators, the differences in skill level were significant. This phenomenon is familiar from countless studies that have found significant differences between subjects. Based on the findings, traditional "expert characteristics", such as length of experience, do not explain the differences and do not guarantee excellent performance.

$RQ2$ addressed the observed differences. In this study, the high performing individuals were significantly faster and more successful when completing the tasks. The top performers were significantly better at predicting their success; they were able to describe in detail what they needed to do and what they needed to ascertain. They also expressed more confidence and effective planning during both the task anticipation and the tasks themselves. Less accomplished subjects were overconfident and optimistic in task anticipations, presumably indicating that they had less of the knowledge needed to make an informed analysis about their performance (Kruger and Dunning, 1999). The findings align with previous research; consistent with this study, experts are able to perform in a way that reliably superior, have a more nuanced problem comprehension, and predict their actions accurately (de Groot, 1978; Ericsson, Krampe, et al., 1993).

*RQ*3 investigated the relationship between the length of professional system administration experience and measured performance. The results were consistent with previous findings, and a relationship between the length of professional experience and measured performance was not found. However, in order to outperform others, subjects had to have more than ten years of experience and intense involvement in computer-related activities both before and during their career. Two of the least accomplished subjects had over 20 years of professional experience as system administrators. The findings align with the core of the deliberate practice approach (Ericsson, Krampe, et al., 1993); experience in itself does not guarantee expertise, but sufficient duration (typically, almost ten years) of intense involvement is required to be among the best performers.

*RQ*4 examined the relationship between the accumulated duration of practice and measured performance. First, it should be noted that summing up hours of any type of ICT-related activity, including work experience, and referring to it as practice – implying that the impact of all types of activities on performance is equal – is strongly against the deliberate practice theory. However, accumulated duration of practice has been found to be an important predictor of individual differences in sports performance (Macnamara, Moreau, et al., 2016) and therefore this measurement aroused our interest. The study by Macnamara, Moreau, and Hambrick (2016) was criticized for adding up all hours of any type of practice, correlating the sum with attained performance, and using the term deliberate practice incorrectly to refer such concept (Ericsson, 2016). Recently, the researchers have debated on the definition of *deliberate practice* and whether the total number of hours of accumulated practice time is a reasonable measure to evaluate correlation between effect of practice on attained performance (Macnamara, Hambrick, and Moreau, 2016; Ericsson, 2016; Ericsson, 2020). To answer *RQ*4, all the professional experience was converted into hours and all the reported computer-related activities both before and during subject's career were added up as accumulated duration of practice. A moderate correlation was found between the measured performance and the accumulated duration of practice, but significant difference was found only between high performers and low performers. The mere accumulated number of hours of practice does not seem to be a predictor of expert performance in system administration. The findings did not either support the "10,000-hour rule" (Gladwell, 2008), since even the low performers had an average of 25,800 hours of practice, but still their performance was very modest compared to others.

*RQ*5 investigated the impact of different forms of practice on performance. The results show that the high-effort training types that meet the boundary conditions of purposeful practice were a prerequisite for being among the top performers. The exact quantity of practice meeting the criteria of purposeful practice cannot be isolated from the reported activities, but the number of activities that were self-assessed as high-effort was significantly greater among high performers. Of the activities, programming dominated since high performers spent significantly more time on it than the other participants. High performers also reported it to require more effort and found it more useful for their professional skills. Research in the field of deliberate practice indicates that the amount of high-quality practice accumulated during individuals' careers is closely related to their attained performance in a wide range of domains (Ericsson, 2008), and the same observation was made in this study.

*RQ*6 studied the required quantity and the best kind of training activities for someone hoping to become a high-performing system administrator. When studying subjects' learning histories, the quantity and quality of programming experience and other high-effort computer-related problem-solving activities was found to be the main difference between 'experts' and less-accomplished participants. These results were consistent with predictions and previous findings. High performers reported spending an average of 33.50 hours per week (4.8 hours per day) in computer-related activities over a long period before their professional career. In contrast, medium performers reported spending an average of 14.5 hours per week (2.1 hours per day) and low performers an average of 9 hours per week (1.3 hours per day). Ericsson, Krampe, and Tesch-Römer (1993) found that the average duration of the violinists at the Music Academy of West Berlin (Hochschule der Kuenste) solo-practice with the violin averaged 24.3 hours of practice per week. It is noteworthy that the top system administrators had spent more time on solitary practice than the top violin students who had the potential for careers as international soloists. In addition to practice alone, the music academy students also took and gave lessons and participated in guided practice sessions; therefore, the total duration of weekly hours was 50. Within the system administrators, a similar trend had emerged during their careers when high performers reported spending an average of 18.5 hours per week (2.6 hours per day), medium performers an average of 6 hours per week (0.9 hours per day), and low performers 5.8 hours per week (0.8 hours per day) in computer-related activities during their free time, on top of their 40-hour work week.

After following the expert-performance approach to answer the research questions, it is essential to notice that typical computer-related activities may not meet all criteria for deliberate or purposeful practice. However, in their meta-analysis, Ericsson and Harwell (2019) found that very few practice activities at all meet all the criteria for the original definition of deliberate practice. It is also questionable whether the actual perceived goal in computer-related activities is the improvement in skills – or accomplishing some other goal, such as configuring a system to function as desired or making a computer program work in a certain way. However, the reported high-effort activities, such as programming, where high performers were engaged in solitary practice does meet the criteria of purposeful practice – such activities have been specifically identified as a typical predictor of superior performance in problem-solving domains such as chess.

A doubt about the generalizability of the results concerns the design of the study. The experiment only measured participants' technical skills, although the system administrator's work includes many other aspects, such as communication, customer service, and teamwork. Why were these aspects not addressed in the experiment? Also, why were all the tasks focused only on creating new features for the system? In real life, an essential part of administrative work is to understand and manage extremely complex systems that have been built over the years. To address these concerns, we can turn to research from the past decades. The pioneering work of selecting critical events was introduced by de Groot (1946 / 1978) in the domain of chess, and ever since, a small collection of representative tasks has been found to be able to capture the essence of expertise in numerous domains. In this thesis, the aim was to make the tasks as representative as possible, keeping in mind that they also had to be operationalizable. However, it must be acknowledged that the experiment focused only on the technical skills of system administrators.

We were looking for and we found a strong correlation between intense involvement in high-effort computer-related activities during a long period and expert performance in system administration. On the basis of prior research, we have a reason to assume that observed purposeful practice with programming is critical factor in the development of system administration skills. However, correlation does not imply causation – the results suggest this, but are not evidence. Nevertheless, we noticed that in the system administration domain, large amounts of "consuming" activities or naïve practice (e.g., gaming and reading technology discussion websites) were not predicting superior performance. Also, the accumulated duration of practice, called *structured practice* (Ericsson and Harwell, 2019), was not found to be the predictive factor of superior performance, although there was a significant difference in high performers' and low performers' accumulated duration of practice. These observations contributes to the ongoing debate on definitions of deliberate practice (Hambrick, Altmann, et al., 2014; Hambrick, Oswald, et al., 2014; Macnamara, Moreau, et al., 2016; Ericsson, 2016), and whether the sum of all hours of practice can accurately predict the superior performance: in this case, it did not.

5.2 Limitations and future ideas

The experimental part of this thesis was carried out in 2011. The essence of ICT systems and system administration work has remained similar since then, but the world has gone online at an ever-increasing speed. The number of Internet users has more than doubled after the experiments were carried out, from 2.2 billion in 2011 to 4.5 billion in 2019 (International Telecommunication Union, 2020). The amount of secure Internet servers (per one million people) has grown 42-fold, from 239 in 2011 to over ten thousand in 2019 (World Bank, 2020). Some aspects, such as the massive growth of cloud-based computing, the increased importance of cybersecurity, and the number of mobile devices have changed radically over the last ten years. In that time, desktop computers have nearly ceased to exist; all network traffic is expected to be secure; and local servers have been replaced by cloud computing services such as Amazon AWS, Microsoft Azure, and Google Cloud.

In the rapidly changing field of technology, the system administrator role is also evolving. Recently, new roles called ”DevOps engineer” and ”site reliability engineer” have provoked debate on the future of the traditional system administrator role (Reddit, 2015). The new roles are combining aspects from both software development and system administration, aiming to shorten the ICT systems development life cycle. Whether the ”sysadmin” role is changing permanently, and if so, how, what are the required core skills, and whether the individual differences can be found are open questions for future research.

6 Conclusions

This thesis contributes to the empirical research on expert performance in system administration and provides the groundwork for further studies on the topic. This study will hopefully help educators, recruiters, students, and professionals understand the prerequisite need for extensive practice to master new aspects of complex system administration skills and acquire in-depth knowledge.

The results of this study, together with research on expertise, show that decades of professional experience, a responsible role in a distinguished organization, extensive higher education, a broad interest in information technology, and considerable time spent on computer-related activities are not enough to make someone an expert system administrator. To be among the best performers in this study, a subject was required to have had intense involvement in high-effort problem-solving activities over a long period; the willingness, courage, and ability to solve problems through experimentation; and a genuine desire to challenge themselves and continuously acquire new knowledge. As previously mentioned, a brilliant system administrator practices their art more intensely than a virtuoso violinist.

Bibliography

Adelson, B. (1984). "When novices surpass experts: The difficulty of a task may increase with expertise." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 10.3, p. 483.

Anderson, E. A. and Patterson, D. (2002). *Researching system administration*. University of California, Berkeley.

Barrett, R., Kandogan, E., Maglio, P. P., Haber, E. M., Takayama, L. A., and Prabaker, M. (2004). "Field studies of computer system administrators: analysis of system management tools and practices". In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. Ed. by J. Herbsleb, pp. 388–395.

Bedard, J. and Chi, M. T. (1992). "Expertise". In: *Current directions in psychological science* 1.4, pp. 135–139.

Bédard, J., Chi, M. T., Graham, L. E., and Shanteau, J. (1993). "Expertise in auditing". In: *Auditing* 12, p. 21.

Bloom, B. S., Sosniak, L. A., et al. (1985). *Developing talent in young people*. Ballantine Books.

Brand-Gruwel, S., Wopereis, I., and Vermetten, Y. (2005). "Information problem solving by experts and novices: Analysis of a complex cognitive skill". In: *Computers in Human Behavior* 21.3, pp. 487–508.

Bureau of Labor Statistics, U.S. Department of Labor (2020). *Occupational Outlook Handbook*. Network and Computer Systems Administrators. URL: https://www.bls.gov/ooh/computer-and-information-technology/network-and-computer-systems-administrators.htm.

Camerer, C. F. and Johnson, E. J. (1991). "The process-performance paradox in expert judgment – How can experts know so much and predict so badly?" In: *Toward a general theory of expertise: Prospects and limits*. Ed. by K. A. Ericsson and J. Smith. Cambridge, UK: Cambridge University Press.

Carroll, J. M. (1997). "Human-computer interaction: psychology as a science of design". In: *Annual review of psychology* 48.1, pp. 61–83.

Charness, N., Krampe, R., and Mayr, U. (1996). "The role of practice and coaching in entrepreneurial skill domains: An international comparison of life-span chess skill acquisition". In: *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games*. Ed. by K. A. Ericsson, pp. 51–80.

Choudhry, N. K., Fletcher, R. H., and Soumerai, S. B. (2005). "Systematic review: the relationship between clinical experience and quality of health care". In: *Annals of Internal medicine* 142.4, pp. 260–273.

Côté, J., Ericsson, K. A., and Law, M. P. (2005). "Tracing the development of athletes using retrospective interview methods: A proposed interview and validation procedure for reported information". In: *Journal of applied sport psychology* 17.1, pp. 1–19.

de Groot, A. D. (1978). "Thought and Choice in Chess". In: *The Hague: Mouton.* Original work published in 1946.

Elstein, A. S., Shulman, L. S., and Sprafka, S. A. (1978). *Medical problem solving: An analysis of clinical reasoning*. Cambridge, MA: Harvard University Press.

Ericsson, K. A. (1996). "The acquisition of expert performance: An introduction to some of the issues". In: *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games*. Ed. by K. A. Ericsson. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc, pp. 1–50.

Ericsson, K. A. (1998). "The scientific study of expert levels of performance: General implications for optimal learning and creativity". In: *High Ability Studies* 9.1, pp. 75–100.

Ericsson, K. A. (2002). "Towards a procedure for eliciting verbal expression of non-verbal experience without reactivity: Interpreting the verbal overshadowing effect within the theoretical framework for protocol analysis". In: *Applied Cognitive Psychology* 16.8, pp. 981–987.

Ericsson, K. A. (2004). "Deliberate practice and the acquisition and maintenance of expert performance in medicine and related domains". In: *Academic medicine* 79.10, S70–S81.

Ericsson, K. A. (2006a). "Protocol analysis and expert thought: Concurrent verbalizations of thinking during experts' performance on representative tasks". In: *The Cambridge Handbook of Expertise and Expert Performance*. Ed. by K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman. Cambridge, UK: Cambridge University Press, pp. 233–241.

Ericsson, K. A. (2006b). "The influence of experience and deliberate practice on the development of superior expert performance". In: *The Cambridge Handbook of Ex-*

pertise and Expert Performance. Ed. by K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman. Cambridge, UK: Cambridge University Press, pp. 683–703.

Ericsson, K. A. (2008). "Deliberate practice and acquisition of expert performance: a general overview". In: *Academic emergency medicine* 15.11, pp. 988–994.

Ericsson, K. A. (2016). "Summing up hours of any type of practice versus identifying optimal practice activities: Commentary on Macnamara, Moreau, & Hambrick (2016)". In: *Perspectives on Psychological Science* 11.3, pp. 351–354.

Ericsson, K. A. (2020). "Towards a science of the acquisition of expert performance in sports: Clarifying the differences between deliberate practice and other types of practice". In: *Journal of Sports Sciences* 38.2, pp. 159–176.

Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. (1993). "The role of deliberate practice in the acquisition of expert performance." In: *Psychological review* 100.3, p. 363.

Ericsson, K. A. and Lehmann, A. C. (1996). "Expert and exceptional performance: Evidence of maximal adaptation to task constraints". In: *Annual review of psychology* 47.1, pp. 273–305.

Ericsson, K. A. and Lehmann, A. C. (1999). "Expertise". In: *Encyclopedia of creativity*. Ed. by S. R. Pritzker and M. A. Runco. Academic Press/Elsevier, pp. 695–707.

Ericsson, K. A. and Pool, R. (2016). *Peak: Secrets from the new science of expertise*. Houghton Mifflin Harcourt.

Ericsson, K. A., Prietula, M. J., and Cokely, E. T. (2007). "The making of an expert". In: *Harvard business review* 85.7/8, p. 114.

Ericsson, K. A. and Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. the MIT Press.

Ericsson, K. A. and Smith, J. (1991). "Prospects and limits of the empirical study of expertise: An introduction". In: *Toward a general theory of expertise: Prospects and limits*. Ed. by K. A. Ericsson and J. Smith. Cambridge, UK: Cambridge University Press.

Ericsson, K. A. and Ward, P. (2007). "Capturing the naturally occurring superior performance of experts in the laboratory: Toward a science of expert and exceptional performance". In: *Current Directions in Psychological Science* 16.6, pp. 346–350.

Ericsson, K. A. and Harwell, K. (2019). "Deliberate practice and proposed limits on the effects of practice on the acquisition of expert performance: Why the original definition matters and recommendations for future research". In: *Frontiers in psychology* 10, p. 2396.

Facer, K., Sutherland, R., Furlong, R., and Furlong, J. (2001). “What’s the point of using computers? The development of young people’s computer expertise in the home”. In: *New Media & Society* 3.2, pp. 199–219.

Fitts, P. M. and Posner, M. I. (1967). “Human performance”. In:

Flora, C. (2005). *The Grandmaster Experiment*. Psychology Today. URL: https://www.psychologytoday.com/intl/articles/200507/the-grandmaster-experiment.

Gabrielsson, A. (1987). “Once again: The theme from Mozart’s piano sonata in A major”. In: *Action and perception in rhythm and music*, pp. 81–103.

Galton, F. (1869). *Hereditary genius: An inquiry into its laws and consequences*. Vol. 27. Macmillan.

Garvie, A. E., Hardie, P., Hunter, R., et al. (1994). *Homer: Odyssey*. Cambridge University Press.

Gergle, D. and Tan, D. S. (2014). “Experimental research in HCI”. In: *Ways of Knowing in HCI*. Ed. by J. S. Olson and W. A. Kellogg. Springer, pp. 191–227.

Gladwell, M. (2008). *Outliers: The story of success*. New York, New York: Little, Brown and Company.

Gobet, F. and Campitelli, G. (2007). “The role of domain-specific practice, handedness, and starting age in chess.” In: *Developmental psychology* 43.1, p. 159.

Hambrick, D. Z., Altmann, E. M., Oswald, F. L., Meinz, E. J., and Gobet, F. (2014). “Facing facts about deliberate practice”. In: *Frontiers in psychology* 5, p. 751.

Hambrick, D. Z., Oswald, F. L., Altmann, E. M., Meinz, E. J., Gobet, F., and Campitelli, G. (2014). “Deliberate practice: Is that all it takes to become an expert?” In: *Intelligence* 45, pp. 34–45.

Hambrick, D. Z. and Tucker-Drob, E. M. (2015). “The genetics of music accomplishment: Evidence for gene–environment correlation and interaction”. In: *Psychonomic bulletin & review* 22.1, pp. 112–120.

Hornbæk, K. et al. (2013). “Some whys and hows of experiments in human–computer interaction”. In: *Foundations and Trends® in Human–Computer Interaction* 5.4, pp. 299–373.

Ilomäki, L. and Rantanen, P. (2007). “Intensive use of ICT in school: Developing differences in students’ ICT expertise”. In: *Computers & Education* 48.1, pp. 119–136.

International Telecommunication Union (2020). *Individuals using the Internet 2011-2019*. URL: https://www.itu.int/en/ITU-D/Statistics/Pages/stat/ (visited on 05/24/2020).

Jeffries, R., Turner, A. A., Polson, P. G., and Atwood, M. E. (1981). "The processes involved in designing software". In: *Cognitive skills and their acquisition* 255, p. 283.

Kaarakainen, M.-T., Kivinen, O., and Vainio, T. (2018). "Performance-based testing for ICT skills assessing: a case study of students and teachers' ICT skills in Finnish schools". In: *Universal Access in the Information Society* 17.2, pp. 349–360.

Keith, N. and Ericsson, K. A. (2007). "A deliberate practice account of typing proficiency in everyday typists". In: *Journal of Experimental Psychology: Applied* 13.3, p. 135.

Kruger, J. and Dunning, D. (1999). "Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments." In: *Journal of personality and social psychology* 77.6, p. 1121.

Kuncicky, D. and Wynn, B. A. (1998). *Educating and training system administrators: A survey*. 4. USENIX Association.

Kyle, D. G. (2013). "Greek Athletic Competitions: The Ancient Olympics and More". In: *A Companion to Sport and Spectacle in Greek and Roman Antiquity*. Ed. by P. Christesen and D. G. Kyle, pp. 17–35.

Limoncelli, T., Hogan, C. J., and Chalup, S. R. (2007). *The practice of system and network administration*. Pearson Education.

Macnamara, B. N., Hambrick, D. Z., and Moreau, D. (2016). "How important is deliberate practice? Reply to Ericsson (2016)". In: *Perspectives on Psychological Science* 11.3, pp. 355–358.

Macnamara, B. N., Hambrick, D. Z., and Oswald, F. L. (2014). "Deliberate practice and performance in music, games, sports, education, and professions: A meta-analysis". In: *Psychological science* 25.8, pp. 1608–1618.

Macnamara, B. N., Moreau, D., and Hambrick, D. Z. (2016). "The relationship between deliberate practice and performance in sports: A meta-analysis". In: *Perspectives on Psychological Science* 11.3, pp. 333–350.

Mayer, R. E. (1997). "From novice to expert". In: *Handbook of human-computer interaction*. Ed. by M. G. Helander, T. K. Landauer, and P. V. Prabhu. Elsevier, pp. 781–795.

Meinz, E. J. and Hambrick, D. Z. (2010). "Deliberate practice is necessary but not sufficient to explain individual differences in piano sight-reading skill: The role of working memory capacity". In: *Psychological science* 21.7, pp. 914–919.

Merriam-Webster (2020). *Expert*. URL: https://www.merriam-webster.com/dictionary/expert (visited on 05/28/2020).

Nemeth, E., Snyder, G., Hein, T. R., Whaley, B., and Mackin, D. (2017). *UNIX and Linux System Administration Handbook (5th Edition)*. 5th. Addison-Wesley Professional.

Newell, K. M. (1991). "Motor skill acquisition". In: *Annual review of psychology* 42.1, pp. 213–237.

OECD (2018). *PISA 2018 Technical report*. OECD Publishing. URL: https://www.oecd.org/pisa/data/pisa2018technicalreport/.

Official Statistics of Finland (2020). *Labour force survey*. URL: http://www.stat.fi/til/tyti/index_en.html (visited on 05/27/2020).

Oulasvirta, A., Wahlström, M., and Ericsson, K. A. (2011). "What does it mean to be good at using a mobile device? An investigation of three levels of experience and skill". In: *International journal of human-computer studies* 69.3, pp. 155–169.

Pennington, N. (1987). "Comprehension strategies in programming". In: *Empirical studies of programmers: second workshop*. Ed. by G. M. Olson, S. Sheppard, and E. Soloway. Ablex Publishing Corp., pp. 100–113.

Plant, E. A., Ericsson, K. A., Hill, L., and Asberg, K. (2005). "Why study time does not predict grade point average across college students: Implications of deliberate practice for academic performance". In: *Contemporary Educational Psychology* 30.1, pp. 96–116.

Plato (2000). *The Republic*. Ed. by G. R. Ferrari and T. Griffith. Original work published ca. 370 B.C.E. Cambridge, UK: Cambridge University Press.

Purchase, H. C. (2012). *Experimental human-computer interaction: a practical guide with visual examples*. Cambridge University Press.

Puustinen, M. and Pulkkinen, L. (2001). "Models of self-regulated learning: A review". In: *Scandinavian Journal of Educational Research* 45.3, pp. 269–286.

Reddit (2015). *Is "DevOps" and/or "SRE" becoming the new "sysadmin?"* URL: https://www.reddit.com/r/sysadmin/comments/3ry602/is_devops_andor_sre_becoming_the_new_sysadmin/ (visited on 05/24/2020).

Schmidt, R. A. and Bjork, R. A. (1992). "New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training". In: *Psychological science* 3.4, pp. 207–218.

Shadish, W., Cook, T. D., and Campbell, D. T. (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Boston, MA.

Simon, H. A. (1973). “The Structure of Ill Structured Problems”. In: *Artificial Intelligence* 4, pp. 181–201.

Simon, H. A. and Chase, W. G. (1973). “Skill in chess”. In: *American Scientist* 61.4, pp. 394–403.

Sloboda, J. A., Davidson, J. W., Howe, M. J., and Moore, D. G. (1996). “The role of practice in the development of performing musicians”. In: *British journal of psychology* 87.2, pp. 287–309.

Sonnentag, S. (2001). “Using and Gaining Experience in Professional”. In: *Linking expertise and naturalistic decision making*, p. 275.

Sonnentag, S., Niessen, C., and Volmer, J. (2006). “Expertise in Software Design”. In: *The Cambridge Handbook of Expertise and Expert Performance*. Ed. by K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman. Cambridge, UK: Cambridge University Press, pp. 373–388.

Takayama, L. and Kandogan, E. (2006). “Trust as an underlying factor of system administrator interface choice”. In: *CHI’06 extended abstracts on Human factors in computing systems*. Ed. by G. Olson, pp. 1391–1396.

USENIX Special Interest Group for Sysadmins (2011). *The 2011 LISA Salary Survey Results*. Tech. rep. URL: https://www.usenix.org/sysadmin-salary-surveys.

Voronkov, A., Martucci, L. A., and Lindskog, S. (2019). “System administrators prefer command line interfaces, don’t they? an exploratory study of firewall interfaces”. In: *Fifteenth Symposium on Usable Privacy and Security 2019)*.

Williams, A. M. and Ward, P. (2003). “Perceptual expertise: Development in sport”. In: *Expert performance in sports: Advances in research on sport expertise*. Ed. by J. L. Starkes and K. A. Ericsson. Human Kinetics Publishers, pp. 219–250.

World Bank (2020). *Secure Internet servers (per 1 million people)*. URL: https://data.worldbank.org/indicator/IT.NET.SECR.P6 (visited on 05/24/2020).

Zimmerman, B. J. (2002). “Becoming a self-regulated learner: An overview”. In: *Theory into practice* 41.2, pp. 64–70.

Appendix A Experiment materials

A.1 Task cards

Figure A.1: Practice task

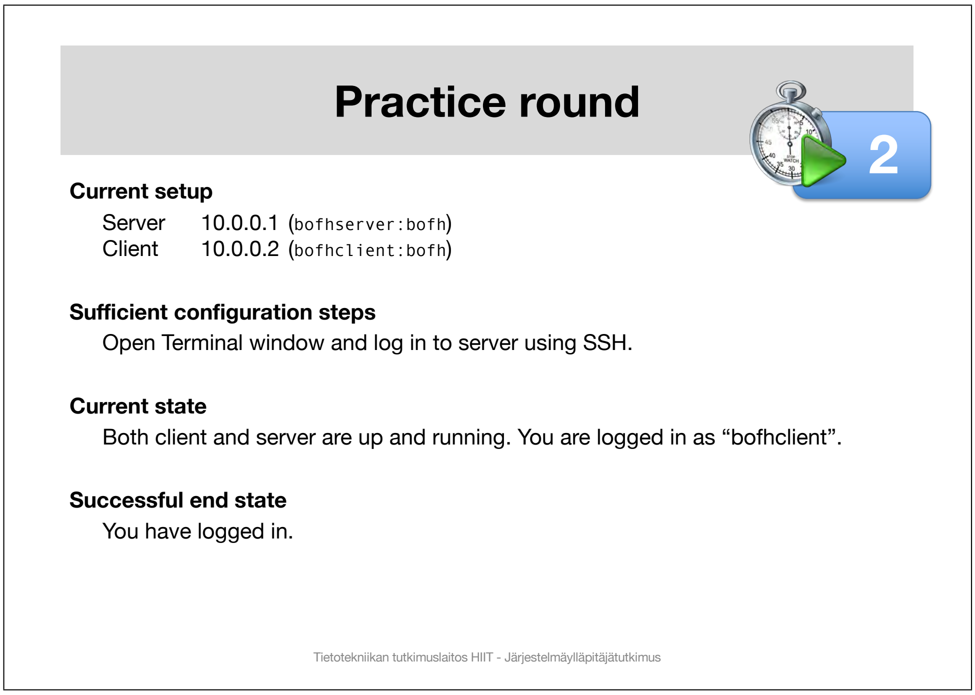


Figure A.2: Task 1

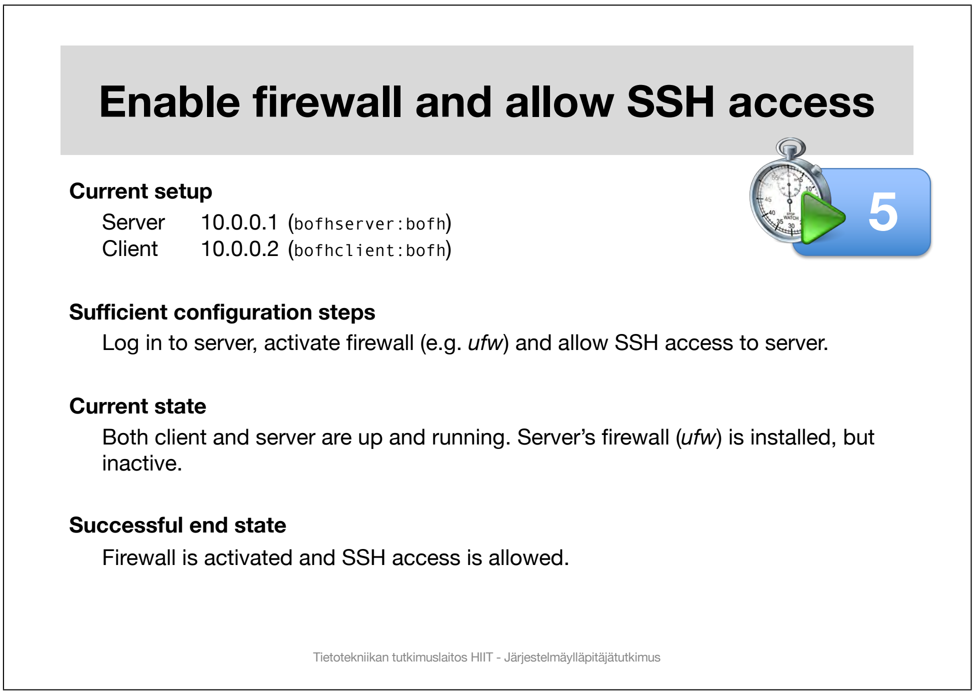


Figure A.3: Task 2

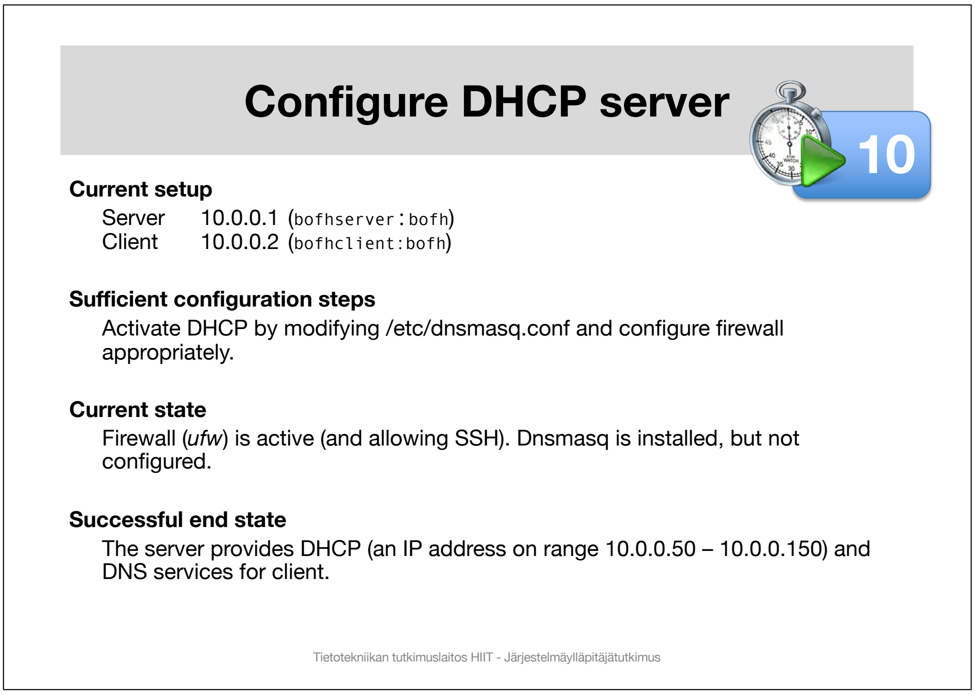


Figure A.4: Task 3

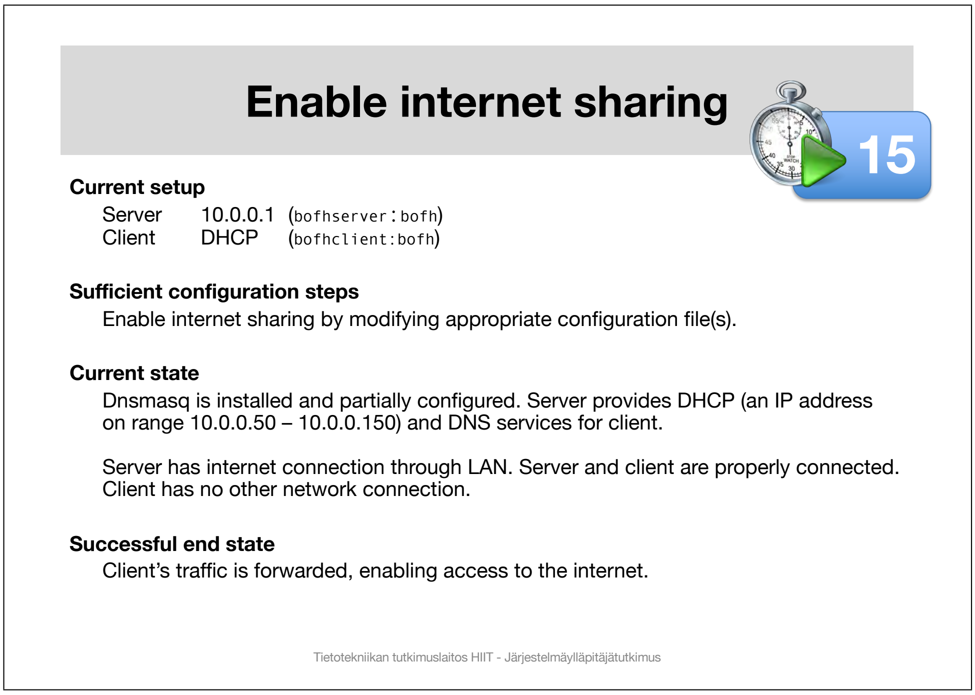


Figure A.5: Task 4

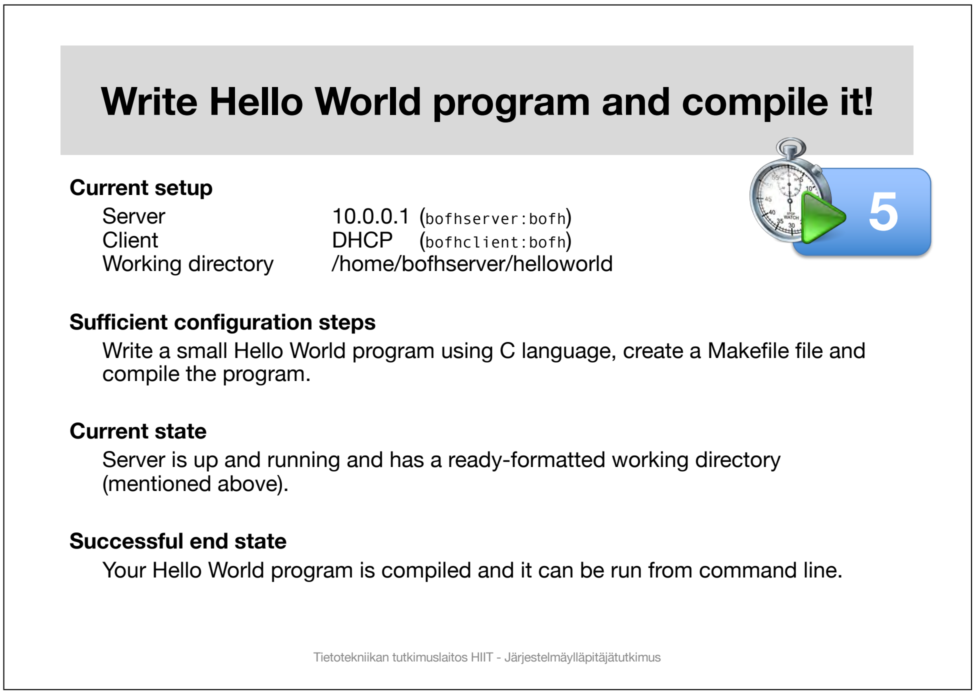


Figure A.6: Task 5

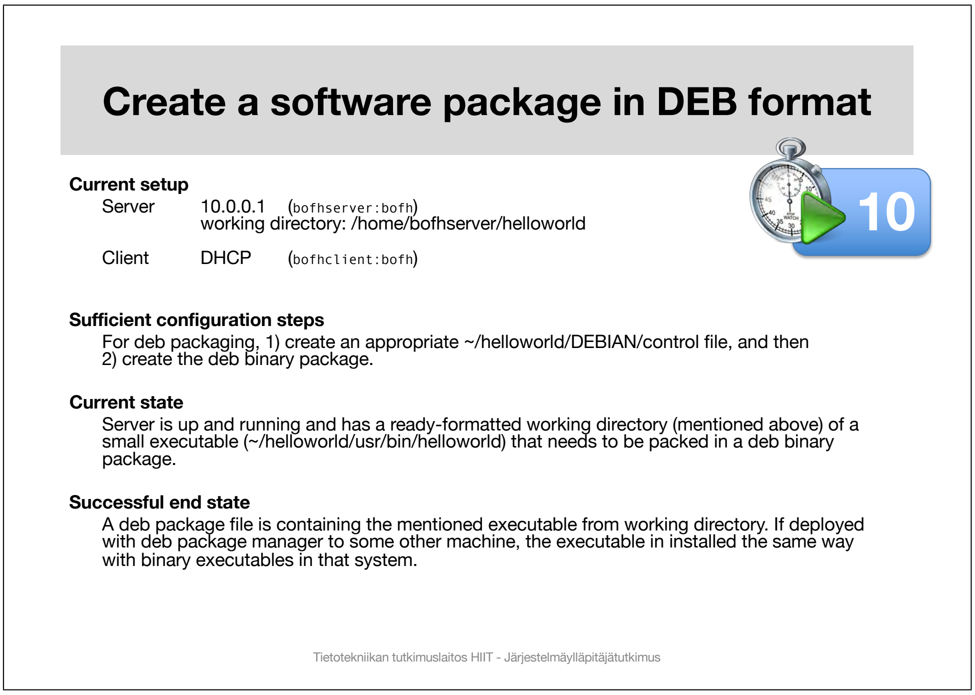


Figure A.7: Task 6

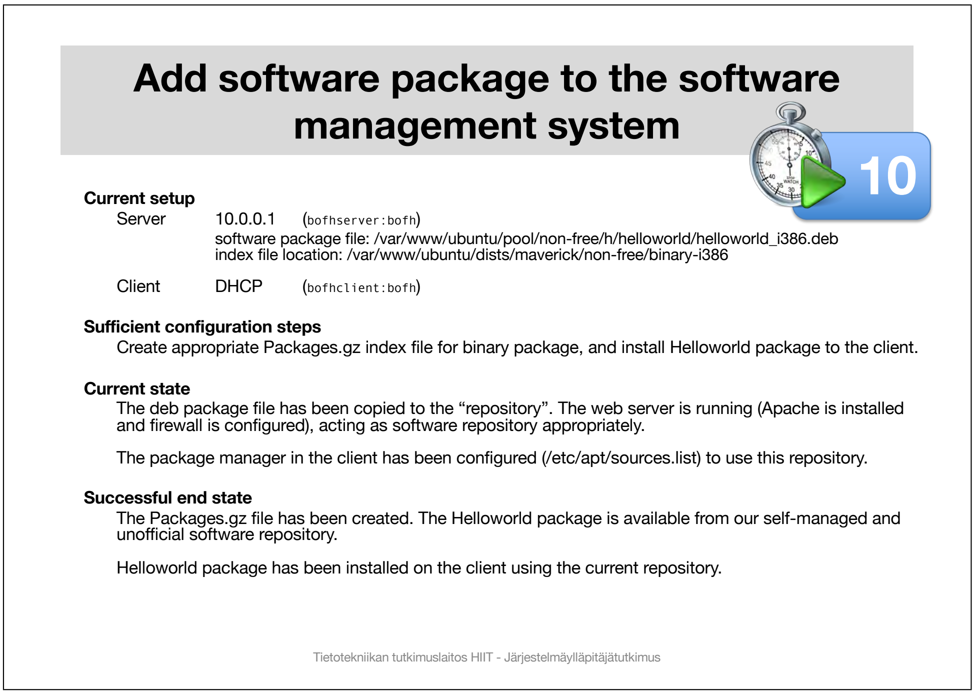


Figure A.8: Task 7

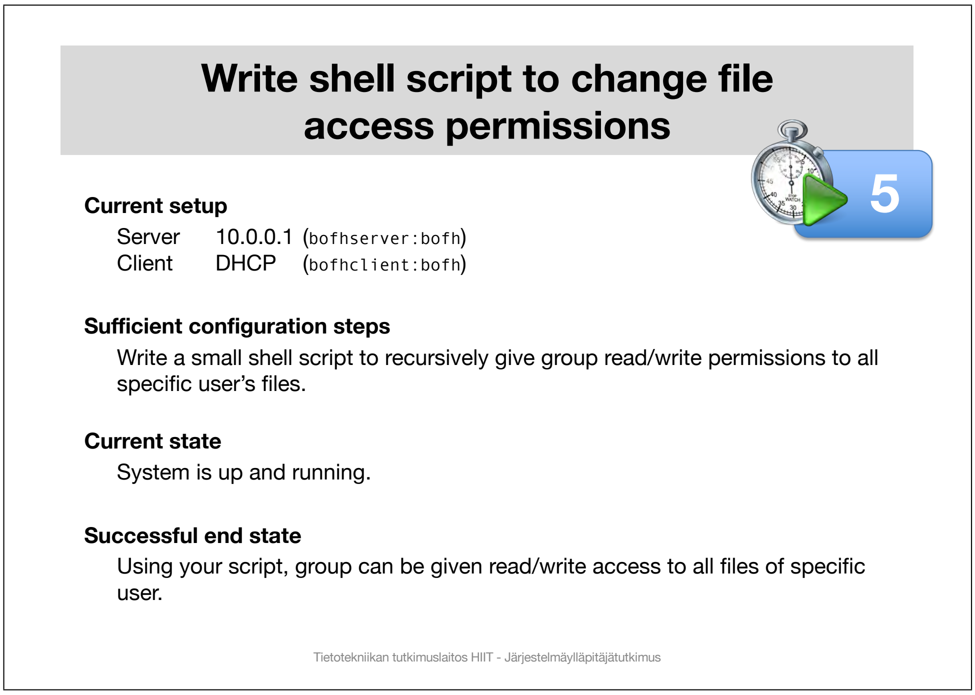


Figure A.9: Task 8

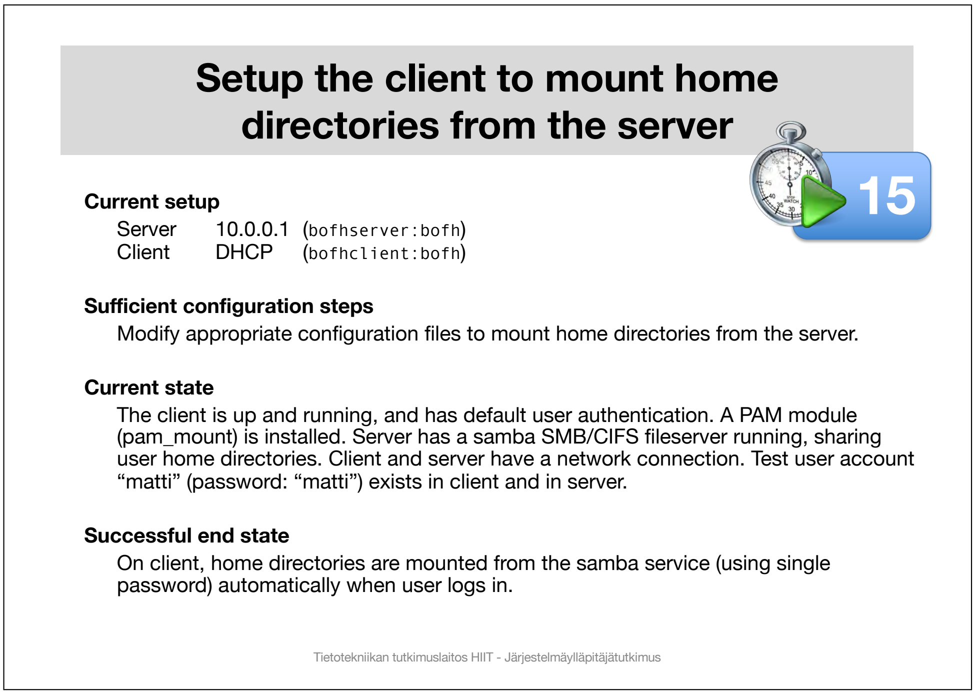


Figure A.10: Task 9

Set up a backup routine for a laptop

15

Current setup

Server	10.0.0.1	(`bofhserver:bofh`)
Client	DHCP	(`bofhclient:bofh`)

Sufficient configuration steps

Write backup script that does two checks before executing backup command (*rsnapshot daily*):

1) checks connection to server (e.g. ping –c 3), and
2) checks that /var/log/run-my-backup.log is older than a day.

Finally, if rsnapshot exits successfully (exit value 0), touch or modify /var/log/run-my-backup.log. Soft-link your script to /etc/network/ip-up.d/ directory.

Current state

The client has users' files in /home directory. Client and server have a working network connection. The data to be backed up can be stored on the server (meaning, you don't have to care about storage space requirements). Backup utility (*rsnapshot*) is installed and configured.

Successful end state

The client's /home directory is backed up automatically every time a network connection to the server is established, but not more often than once a day.

Tietotekniikan tutkimuslaitos HIIT - Järjestelmäylläpitäjätutkimus

A.2 Participant information sheet

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

TUTKIMUSTIEDOTE **1 (2)**

11.7.2011

Järjestelmäylläpitäjätutkimus

Tutkimuksen toteuttaja ja tavoitteet

Järjestelmäylläpitäjätutkimuksen tarkoituksena on tuottaa ymmärrystä tietojärjestelmäylläpitäjien asiantuntijuuteen ja taitotasoon vaikuttavista tekijöistä. Tutkimuksen suorittaa Aalto-yliopiston ja Helsingin yliopiston yhteinen tietotekniikan tutkimuslaitos HIIT.

Mitä tutkimuksessa tapahtuu

Tutkimus jakautuu kahteen vaiheeseen. Ensimmäisessä vaiheessa tutkimushenkilöt suorittavat tehtävänannon mukaisia Linux-ylläpitotehtäviä tarkoitusta varten asennetun järjestelmän avulla. Ylläpitotehtävät suoritetaan tutkimushenkilön omassa työympäristössä tai tietotekniikan tutkimuslaitos HIIT:in tiloissa. Tutkimuksen toisessa vaiheessa tutkimushenkilöitä haastatellaan heidän tietotekniseen osaamistasoon vaikuttavien taustatekijöiden kartoittamiseksi. Tutkimuksen molemmat vaiheet videoidaan.

Mitä tutkimukseen osallistuminen vaatii tutkimushenkilöltä

Tutkimukseen osallistuminen vie tutkimushenkilöiden aikaa ylläpitotehtävävaiheen osalta noin kaksi (2) tuntia ja myöhemmin suoritettavan haastattelun osalta noin yhden (1) tunnin. Tutkimuksessa tehtävät ylläpitotehtävät suoritetaan HIIT:in tarkoitusta varten asentamia laitteita ja ohjelmistoja käyttäen eikä tutkimushenkilön tai tämän työnantajan tietojärjestelmiä käytetä tutkimuksessa millään tavoin.

Kuinka tietoja käsitellään

Tutkimuksessa kertyvä aineistot, kuten videotallenteet, lokitiedostot ja haastatteluvastaukset, ovat luottamuksellista tietoa. Tutkijoilla on vaitiolo- ja salassapitovelvollisuus, eikä tutkimusta varten kerttyjä tietoja luovuteta kolmanislle osapuolille.

TUTKIMUSTIEDOTE **2 (2)**

11.7.2011

Korvaus osallistumisesta

Tutkimushenkilöille maksetaan 100 euron palkkio osallistumisestaan.

Tutkimushenkilöt ovat osallistumisensa kautta myös mukana tuottamassa uutta ymmärrystä tietoteknisellä alalla asiantuntijuuteen vaikuttavista tekijöistä ja heille kerrotaan tutkimuksen tuloksista niiden valmistuttua.

Tutkimuksen keskeyttäminen

Tutkimushenkilöillä on oikeus keskeyttää osallistumisensa tutkimukseen koska tahansa ja syytä ilmoittamatta.

Yhteystiedot

Tutkimuksen vastuullinen johtaja:
Antti Oulasvirta
antti.oulasvirta@hiit.fi
+358 50 3841561

Tutkimusrekisterinpitäjä:
Tietotekniikan tutkimuslaitos HIIT
PO Box 19215, 00076 Aalto

A.3 Consent form

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

SUOSTUMUS JÄRJESTELMÄYLLÄPITÄJÄTUTKIMUKSEEN

Minua on pyydetty osallistumaan järjestelmäylläpitäjätutkimukseen.

Olen saanut, lukenut ja ymmärtänyt tutkimuksesta kertovan tutkimustiedotteen. Tiedotteesta olen saanut riittävän selvityksen tutkimuksesta ja sen yhteydessä suoritettavasta tietojen keräämisestä, käsittelystä ja luovuttamisesta. Tiedotteen sisältö on kerrottu minulle myös suullisesti ja olen saanut riittävän vastauksen kaikkiin tutkimusta koskeviin kysymyksiini.

Tiedot antoi ____ ____ / ____ / _______.

Kaikki tutkimuksen aikana kerättävät tiedot käsitellään luottamuksellisina. Kerättyjä tietoja käsittelevät vain Tuomas Husu ja Antti Oulasvirta Tietotekniikan tutkimuslaitokselta. Tietoja ei koskaan luovuteta kolmansille osapuolille ilman tutkimushenkilön kirjallista lupaa. Tutkimuksesta julkaistavat tieteelliset julkaisut eivät sisällä tietoja, joiden perusteella tutkimushenkilöt tai heidän työnantajansa voisi yksilöidä.

Ymmärrän, että osallistumiseni tähän tutkimukseen on täysin vapaaehtoista. Minulla on oikeus milloin tahansa tutkimuksen aikana ja syytä ilmoittamatta keskeyttää tutkimukseen osallistuminen.

Allekirjoituksellani vahvistan osallistumisen tähän tutkimukseen ja suostun vapaaehtoisesti tutkimushenkilöksi.

_________________________________ ___________________

Allekirjoitus ja nimenselvennys Päiväys

Tästä lomakkeesta on kaksi kopiota: toinen tutkimushenkilöllä ja toinen Tietotekniikan tutkimuslaitoksella.

Helsinki Institute for Information Technology HIIT
Tietotekniikan tutkimuslaitos HIIT
Forskningsinstitutet för informationsteknologi HIIT
www.hiit.fi

Aalto University

UNIVERSITY OF HELSINKI

A.4 Task representativeness questionnaire

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

Tehtäväkysely

#	Tehtävä	Toistuvuus (krt/kk)	Edustavuus
1	Enable firewall and allow SSH access		1 2 3 4 5
2	Configure DHCP server		1 2 3 4 5
3	Enable internet sharing		1 2 3 4 5
4	Write and compile Hello World		1 2 3 4 5
5	Software package in DEB format		1 2 3 4 5
6	Add DEB package to software repository		1 2 3 4 5
7	File permissions script		1 2 3 4 5
8	Mount home directories		1 2 3 4 5
9	Backup script for client		1 2 3 4 5

Toistuvuus = Vastaavan kaltainen tehtävä toistuu työssäni arviolta näin monta kertaa kuukausittain

Edustavuus = Tehtävä edustaa tyypillisiä työtehtäviäni 1 = ei lainkaan, 5 = erittäin hyvin.

Appendix B Coding manual

Behavioral categories

Category	Explanation	Example
Reached sub-goal	Predefined sub-goal reached	One or more commands (e.g., `ufw allow 53/tcp`) have been executed and the system has been set to the state required by the sub-goal
Directly useful action	Acting on an optimal solution path	E.g., `ufw allow 67/tcp` to successfully allow DHCP traffic
Useless action	An action that changes system state, but does not help on task completion	Adding a new rule to firewall although the relevant port is already open
Harmful action	An action that needs to be undone to complete the task	Enabling a firewall over SSH before allowing port 22/tcp, blocking the user out
Information search: system	Reading documentation inside the system	”How do I use `mtime` and `ctime` parameters with `find?`”, `man find`
Information search: external	Using Internet search engines	”Let’s google the syntax, `packages control file syntax`...”
Testing and debugging	Constructive testing of the solution	Proving that some piece of funtionality works: ”Everything seems to be working now, but let’s validate with `dhclient`”

Table B.1: Coding manual: behavioral categories.

Cognitive categories

Category	**Description**	**Example**
Planning and anticipation	Expressing planned actions to solve the task	”And now, when I restart `dnsmasq` the DHCP server will start offering IP addresses from range 10.0.0.50 to 10.0.0.150, but before that we need to...“
Confidence	Expressing confidence	”Ok, now I understand. This `ufw` tool is just a frontend for `iptables`. . . ”
Unconfidence	Expressing unconfidence	”I don’t understand, I have no idea how to get this working...”

Table B.2: Coding manual: cognitive categories.

Appendix C Task performance results

C.1 Task success per subject

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	%	$\sum$
P1	1	1	1	1	0	1	1	1	1	89	8
P2	1	1	0	1	0	1	1	0	0	56	5
P3	1	1	1	1	1	1	1	0	1	89	8
P4	1	0	0	0	0	0	1	0	0	22	2
P5	0	0	0	0	1	0	1	0	1	33	3
P6	1	0	0	0	0	0	0	0	0	11	1
P7	1	1	0	1	1	1	1	1	1	89	8
P8	0	1	1	1	1	1	1	1	1	89	8
P9	1	1	1	1	1	1	1	0	1	89	8
P10	1	0	0	1	0	0	0	0	0	22	2
P11	1	1	0	1	1	0	1	0	0	56	5
P12	0	0	1	1	1	0	0	0	0	33	3
P13	1	0	0	1	0	1	0	0	0	33	3
P14	1	0	0	0	0	0	0	0	0	11	1
P15	0	1	0	1	1	1	1	0	0	56	5
P16	1	0	0	1	1	1	1	0	1	67	6
P17	0	1	1	1	0	1	1	1	1	78	7
P18	1	0	0	0	0	0	0	0	0	11	1
P19	1	0	0	1	1	1	0	0	0	44	4
P20	0	0	1	1	0	0	1	1	1	56	5
%	70	45	35	75	50	55	65	25	45		
$\sum$	14	9	7	15	10	11	13	5	9		

Note. $\sum$ = total.

Table C.1: Task success per subject (1 = success, 0 = failure).

C.2 Anticipated success rate per subject

	T1	T2	T3	T4	T5	T6	T7	T8	T9	$\bar{x}$	M	SD
P1	0.9	0.9	0.9	1.0	0.75	0.7	0.9	0.8	1.0	0.87	0.90	0.10
P2	0.9	0.8	0.9	0.8	0.7	0.7	0.9	0.6	0.7	0.69	0.70	0.28
P3	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.8	0.95	0.88	0.90	0.05
P4	0.5	0.5	0.2	0.2	0.05	0	0.05	0.05	0	0.17	0.05	0.20
P5	0.75	0.3	0.2	0.2	0.05	0.1	0.5	0.1	0.75	0.33	0.20	0.27
P6	0.5	0.75	0.4	0.8	0.4	0.2	0.9	0.75	0.7	0.60	0.70	0.23
P7	0.9	0.9	0.5	0.6	0.2	0.2	0.8	0.1	0.8	0.56	0.60	0.32
P8	0.8	0.8	0.8	1.0	1.0	1.0	1.0	1.0	1.0	0.93	1.00	0.10
P9	1.0	1.0	1.0	1.0	0.8	1.0	1.0	1.0	1.0	0.98	1.00	0.07
P10	1.0	1.0	1.0	0.75	0.3	0.3	0.5	1.0	0	0.65	0.75	0.39
P11	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.50	0.50	0.00
P12	1.0	0.9	1.0	0.5	0.5	0.5	0	0.4	0.5	0.59	0.50	0.33
P13	0.3	0.35	0.4	0.2	0.5	0.6	0.7	0.2	0.3	0.39	0.35	0.17
P14	0.85	0.55	0.3	0.5	0.45	0.25	0.5	0.3	0	0.41	0.45	0.24
P15	0.75	0.6	0.6	0.5	0.6	0.7	0.7	0.6	0.7	0.64	0.60	0.08
P16	1.0	0.8	1.0	1.0	0.4	0.9	1.0	0.85	0.95	0.88	0.95	0.19
P17	1.0	0.9	1.0	1.0	0.7	0.75	1.0	1.0	1.0	0.93	1.00	0.12
P18	0.8	0.6	0.7	0.2	0.2	0.7	0.8	0.7	0.5	0.58	0.70	0.23
P19	0.9	0.85	0.8	0.85	0.7	0.7	0.4	0.5	0.4	0.68	0.70	0.20
P20	0.5	0.6	0.5	0.5	0.01	0.05	0.75	0.01	0.5	0.38	0.50	0.28
$\bar{x}$	0.79	0.72	0.68	0.65	0.49	0.53	0.69	0.56	0.61			
M	0.88	0.80	0.75	0.68	0.50	0.65	0.78	0.60	0.70			
SD	0.21	0.21	0.29	0.30	0.29	0.32	0.30	0.34	0.34			

Note. $\bar{x}$ = mean, M = median, SD = standard deviation.

Table C.2: Anticipated success rate on scale 0 (0%) to 1 (100%).

C.3 Task completion times per subject

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	$\sum$	$\bar{x}$	M	SD
P1	0	0	0	150		0	75	10	120	355	44	5	62
P2	180	0		180		30	30			420	84	30	88
P3	150	0	120	60	0	330	60		120	840	105	90	106
P4	20						0			20	10	10	14
P5					0		60		0	60	20	0	35
P6	180									180	180	180	
P7	60	0		0	60	0	0	270	0	390	49	0	93
P8		270	390	180	300	360	60	240	0	1800	225	255	138
P9	285	440	570	200	430	300	180		450	2855	357	365	137
P10	0			0						0	0	0	0
P11	30	0		0	150		30			210	42	30	62
P12			450	0	90					540	180	90	238
P13	30			0		0				30	10	0	17
P14	135									135	135	135	
P15		300		30	75	0	60			465	93	60	119
P16	0			165	120	0	0		0	285	48	0	75
P17		120	420	0		60	180	15	60	855	122	60	145
P18	0									0	0	0	
P19	165			90	90	70				415	104	90	42
P20			120	0			0	420	105	645	129	105	172
$\sum$	1235	1130	2070	1055	1315	1150	735	946	855				
$\bar{x}$	88	126	296	70	132	105	57	189	95				
M	45	0	390	30	90	30	60	240	60				
SD	92	169	213	81	135	148	61	179	143				

Note. Empty cells represent failures.
$\sum$ = total, $\bar{x}$ = mean, M = median, SD = standard deviation.

Table C.3: Task completion times: time left upon completion (seconds).

C.4 Perceived task representativeness per subject

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	$\bar{x}$	M	SD
P1	5	4	3	4	4	5	5	2	4	4.00	4	1.00
P2	4	1	1	1	1	1	5	1	4	2.11	1	1.69
P3	5	4	3	4	5	5	5	4	5	4.44	5	0.73
P4	4	2	2	1	2	3	5	3	4	2.89	3	1.27
P5	4	3	1	1	1	2	4	3	4	2.56	3	1.33
P6	5	3	1	3	4	4	5	5	5	3.89	4	1.36
P7	4	2	1	3	2	1	5	2	3	2.56	2	1.33
P8	4	3	3	5	1	1	3	2	4	2.89	3	1.36
P9	4	4	2	3	2	3	5	4	5	3.56	4	1.13
P10	5	5	4	1	1	1	5	5	5	3.56	5	1.94
P11	2	2	1	1	1	1	2	1	2	1.44	1	0.53
P12	3	4	4	2	2	2	4	3	4	3.11	3	0.93
P13	4	4	2	1	2	2	5	3	3	2.89	3	1.27
P14	4	2	2	1	3	3	3	3	3	2.67	3	0.87
P15	4	3	3	4	4	4	4	4	4	3.78	4	0.44
P16	4	1	2	5	2	1	4	2	3	2.67	2	1.41
P17	5	4	4	2	1	1	5	5	5	3.56	4	1.74
P18	5	4	1	1	1	2	5	4	4	3.00	4	1.73
P19	4	4	3	5	1	2	4	3	5	3.44	4	1.33
P20	4	3	1	1	3	3	3	3	3	2.67	3	1.00
$\bar{x}$	4.15	3.10	2.20	2.45	2.15	2.35	4.30	3.10	3.95			
M	4	3	2	2	2	2	5	3	4			
SD	0.75	1.12	1.11	1.57	1.27	1.35	0.92	1.21	0.89			

Note. $\bar{x}$ = mean, M = median, SD = standard deviation.

Table C.4: ”Task represents my typical job duties” (1 = strongly disagree, 5 = strongly agree).